



Aalto-yliopisto  
Insinöörیتieteiden  
korkeakoulu

Arsi Juote

## Telemetriatiedon karttasovitusalgoritmi katujen kunnossapidon tilannekuvaa varten

Diplomityö  
Rakennetun ympäristön laitos  
Insinöörیتieteiden korkeakoulu  
Aalto-yliopisto

Espoossa 28.11.2016

Arsi Juote

Valvoja: Professori Kirsi Virrantaus  
Ohjaajat: DI Kimmo Junttila, DI Sami Rapo

---

**Tekijä** Arsi Juote

---

**Työn nimi** Telemetria tiedon karttasovitus algoritmi katujen kunnossapidon tilannekuva varten

---

**Koulutusohjelma** Geomatiikka

---

**Pää-/sivuaine** Geoinformatiikka

**Koodi** M3002

---

**Työn valvoja** Professori Kirsi Virrantaus

---

**Työn ohjaajat** DI Kimmo Junttila, DI Sami Rapo, Vantaan kaupunki

---

**Päivämäärä** 28.11.2016

**Sivumäärä** 88

**Kieli** Suomi

---

## Tiivistelmä

Tämän diplomityön tavoitteena oli edesauttaa Vantaan kaupungin satelliittipaikannukseen perustuvan ajoneuvojen telemetriajärjestelmän tuottaman liikeratadatan käyttöä soveltamalla se tunnettuun digitaaliseen tieverkkoon. Tällaista ohjelmallista toimenpidettä kutsutaan karttasovituksiksi. Tutkimuksessa keskityttiin katujen talvikunnossapitoa tekevien ajoneuvojen liikeratoihin.

Diplomityön tutkimukseen kuului kirjallisuustutkimus sekä konstrukttiivinen osio, jossa toteutettiin kaksi eri menetelmiin perustuvaa algoritmia ja niitä vertailtiin empiirisiin kokein. Teoreettinen pohja tutkimuksen tekemiselle luotiin perehtymällä liikennetelematiikkaan, satelliittipaikannukseen sekä karttasovitukseen. Kirjallisuustutkimuksessa perehdyttiin kattavasti eri sovelluksia varten kehitettyihin karttasovitusalgoritmeihin, ja tunnistettiin nykyisin parhaana pidetyt algoritmit ongelman ratkaisemiseksi.

Työssä toteutettiin kirjallisuudessa hyvin tunnettu Markovin piilomalliin pohjautuva algoritmi ja uudenlainen sumeaan logiikkaan ja reitinhakuun pohjautuva karttasovitusalgoritmi. Kehitetty sumeaan logiikan algoritmi otti huomioon havaintodatan ja tieverkon geometriset, topologiset ja temporaaliset ominaisuudet muodostamalla pisteytyksen sumeassa järjestelmässä kuudesta eri muuttujasta lukuisille erilaisille reittikandidaateille, joista pystyttiin lopuksi muodostamaan globaalisti optimaalinen reitti.

Algoritmien sovitustarkkuutta sekä laskennallista suorituskkyä vertailtiin kerätyn referenssiaineiston avulla. Tutkimuksessa sumeaan logiikkaan pohjautuva karttasovitus todettiin sovitustarkkuudeltaan paremmaksi kuin Markovin piilomalliin pohjautuva algoritmi. Sumeaan logiikan algoritmi integroitiin osaksi Vantaan paikkatietoinfrastruktuuria luomalla geoproessointiketjut, joiden tuloksena havaintodatasta luotiin jatkuvasti päivittyvä karttasovitetun reittien tietokanta katujen kunnossapitotilanteesta. Tietokannan sisältö julkaistiin WMS- ja WFS-paikkatietorajapinnoissa sekä REST-kyselyrajapinnassa tiedon jatkohyödyntämistä varten.

---

**Avainsanat** Liikennetelematiikka, telemetria, karttasovitusalgoritmi, sumea logiikka, Markovin piilomalli

---

|   |                           |                         |
|---|---------------------------|-------------------------|
| <b>Author</b> Arsi Juote  |                           |                         |
| <b>Title of thesis</b> Map matching algorithm for road maintenance situation awareness utilizing telemetry data |                           |                         |
| <b>Degree programme</b> Degree Programme in Geomatics   |                           |                         |
| <b>Major/minor</b> Geoinformatics   |                           | <b>Code</b> M3002       |
| <b>Thesis supervisor</b> Professor Kirsi Virrantaus   |                           |                         |
| <b>Thesis advisor(s)</b> Kimmo Junttila MSc, Sami Rapo MSc, City of Vantaa                                      |                           |                         |
| <b>Date</b> 28.11.2016  | <b>Number of pages</b> 88 | <b>Language</b> Finnish |

## Abstract

The purpose of this master's thesis was to help better utilize data produced by the telemetry system installed on the vehicle fleet of city of Vantaa by matching the GNSS-positioned data to a digital road network. Algorithmic matching of trajectory data to a digital road network is known as map matching. This study focused on vehicles performing duties related to winter maintenance of the road network.

The study involved a literature study and a constructive part where two map matching algorithms based on different methods were implemented and compared by empiric tests. Theoretical background to perform the study was acquired by studying the existing literature on transportation telematics, satellite positioning and map matching. The literature study involved a thorough review of existing map matching algorithms and methods related to different applications. Best practices and algorithms to suit the needs of this study were identified during this phase.

A well-known map matching algorithm based on hidden Markov model was implemented in the study as well as a novel map matching algorithm based on fuzzy logic and path finding. The developed fuzzy logic algorithm evaluated the geometric, topological and temporal properties of the trajectory data as well as the digital road network based on six input parameters. A fuzzy inference system produced a scoring based on the inputs for multiple possible candidate paths. Globally optimal final route was produced by modeling the candidate paths as a graph.

Map matching accuracy and computational performance of the implemented algorithms was evaluated by comparison with manually labeled reference data. The fuzzy logic algorithm outperformed the hidden Markov model algorithm in map matching accuracy and slightly outperformed it in computational performance. The fuzzy logic algorithm was integrated into the GIS infrastructure of city of Vantaa by creating a geoprocessing chain that produces continuously updating database of map matched paths including attribute data related to maintenance telemetry. The data was published as WMS and WFS GIS web services as well as a RESTful API that enables querying the database.

---

**Keywords** Transportation telematics, telemetry, map matching algorithm, fuzzy logic, hidden Markov model

---

## Alkusanat

*Tämä diplomityö on tehty Vantaan kaupungin mittausosastolle. Työn aihepiirin eli telemetriadiedon käsittelyn pariin johdattamisesta kiitokset kuuluvat kaupungin paikkatiedoista vastaavalle ICT-projektipäällikölle Sami Rapolle. Ehdotin ilmenneiden karttasovittukseen liittyvien haasteiden ratkaisuksi diplomityön tekemistä kehittyneistä menetelmistä, johon työnantajani suostui.*

*Haluan kiittää työni valvojaa professori Kirsi Virrantausta sekä työn ohjaajina toimineita Kimmo Junttilaa ja Sami Rapoa. Lopuksi haluan kiittää Vantaan kaupungin mittausosastoa mielenkiintoisten ja haastavien työtehtävien tarjoamisesta aina ensimmäisestä opiskeluvuodestani lähtien.*

Vantaalla marraskuussa 2016

Arsi Juote

# Sisällysluettelo

|   |    |
|---|----|
| Tiivistelmä   |    |
| Abstract  |    |
| Alkusanat   |    |
| Sisällysluettelo .....  | 5  |
| Lyhenteet .....   | 7  |
| Keskeiset käsitteet .....   | 8  |
| 1 Johdanto .....  | 9  |
| 1.1 Työn taustat .....  | 9  |
| 1.2 Työn tavoitteet .....   | 9  |
| 1.3 Tutkimusmenetelmät .....                                      | 10 |
| 1.4 Työn rakenne .....  | 10 |
| 2 Liikennetelematiikka .....                                      | 12 |
| 2.1 Sovellukset .....   | 12 |
| 2.2 Yksityisyydensuoja .....                                      | 15 |
| 2.3 Kehitettävän sovelluksen erityishaasteet .....                | 16 |
| 3 Paikannus .....   | 18 |
| 3.1 Satelliittipaikannus .....                                    | 18 |
| 3.1.1 Sijainnin määrittäminen .....                               | 19 |
| 3.1.2 Tarkkuus ja virhelähteet .....                              | 20 |
| 3.1.3 Differentiaalinen paikannus .....                           | 22 |
| 3.1.4 Tarkemmat menetelmät .....                                  | 22 |
| 3.2 Merkintälasku .....   | 23 |
| 4 Karttasovitus .....   | 24 |
| 4.1 Kirjallisuuskatsaus algoritmeista .....                       | 26 |
| 4.2 Digitaalisen tieverkon merkitys karttasovituksessa .....      | 33 |
| 4.2.1 Paikkatiedon laatu .....                                    | 34 |
| 4.2.2 Lähimmän naapurin haku .....                                | 35 |
| 4.2.3 Lyhyimmän reitin ongelma .....                              | 36 |
| 4.3 Yhteenveto .....  | 38 |
| 5 Toteutettavien algoritmien teoriaa .....                        | 40 |
| 5.1 Markov-mallit .....   | 40 |
| 5.2 Sumea logiikka .....  | 42 |
| 5.2.1 Mamdani-tyypin sumeat järjestelmät .....                    | 43 |
| 5.2.2 Sugeno-tyypin sumeat järjestelmät .....                     | 45 |
| 6 Datan kerääminen .....  | 46 |
| 6.1 Järjestelmän kuvaus .....                                     | 46 |
| 6.2 Tilatiedon automaattinen määrittäminen .....                  | 47 |
| 6.3 Järjestelmän havaintopisteet .....                            | 49 |
| 6.4 Tieverkko .....   | 50 |
| 7 Karttasovitusalgoritmien toteutus .....                         | 53 |
| 7.1 HMM-karttasovitus .....                                       | 53 |
| 7.1.1 Prosessin kulku .....                                       | 53 |
| 7.1.2 Emissio- ja transitiotodennäköisyyksien määrittäminen ..... | 54 |
| 7.1.3 Algoritmin testaus .....                                    | 56 |
| 7.1.4 Erikoistapaukset ja viritykset .....                        | 57 |
| 7.2 Sumea karttasovitus .....                                     | 60 |

|       |   |    |
|-------|---|----|
| 7.3   | Algoritmien vertailu .....                  | 67 |
| 7.3.1 | Referenssiaineisto .....                    | 69 |
| 7.3.2 | Vertailun tulokset.....                     | 71 |
| 7.3.3 | Laskennallisen suorituskyvyn vertailu ..... | 72 |
| 8     | Karttasovituksen hyödyntäminen .....        | 74 |
| 9     | Johtopäätökset .....                        | 76 |
| 10    | Yhteenveto .....                            | 79 |
| 11    | Lähteluettelo .....                         | 80 |
|       | Liiteluettelo .....                         | 88 |

## Lyhenteet

|      |   |
|------|---|
| CRF  | Conditional Random Field  |
| DR   | Dead Reckoning eli merkintälasku tarkoittaa paikantamista suhteessa tunnettuun lähtöpisteeseen kuljetun matkan ja suunnan perusteella   |
| DST  | Dempster-Shafer-Theory, matemaattinen malli, jolla voidaan arvioida uskomusta tukevien todisteiden uskottavuutta  |
| DOP  | Dilution of Precision, nimitys satelliittipaikannuksen epätarkkuuksille, jotka johtuvat satelliittien heikosta näkyvyydestä tai huonosta asemasta taivaankannella suhteessa vastaanottimeen |
| FFS  | FME Feature Store   |
| FIS  | Fuzzy Inference System (suom. sumea järjestelmä)  |
| GNSS | Global Navigation Satellite System  |
| GPS  | Global Positioning System, Yhdysvaltain puolustusministeriön satelliittipaikannusjärjestelmä  |
| HMM  | Hidden Markov Model (suom. Markovin piilomalli)   |
| KLV  | Kevyen liikenteen väylä   |
| REST | Representational State Transfer, tyyli toteuttaa rajapintoja tietokoneiden väliseen viestintään internetin välityksellä noudattaen WWW:n (World Wide Web) periaatteita                      |
| RFID | Radio-Frequency Identification, radioaaltoihin perustuva tunnistemenetelmä.   |
| RTK  | Real Time Kinematic (mittaus), kahteen vastaanottimeen perustuva tarkka satelliittipaikannusmenetelmä   |
| VRS  | Virtual Reference Station, RTK-mittaukseen liittyvä virtuaalinen mittausasema   |
| WFS  | Web Feature Service, rajapintapalvelu paikkatiedon välittämiseen vektorimuodossa  |
| WMS  | Web Map Service, rajapintapalvelu paikkatiedon välittämiseen rasterimuodossa  |
| XML  | Extensible Markup Language  |

## Keskeiset käsitteet

**Paikannus** tarkoittaa kohteen sijainnin määrittämistä suhteessa vertausjärjestelmään, kuten koordinaattijärjestelmään esimerkiksi satelliittipaikannusjärjestelmiä hyödyntäen. Paikannustieto voidaan tallentaa maanpinnan pistettä kuvaavina koordinaatteina.

**Liikeratatieto** tarkoittaa liikkuvasta kohteesta tehtyjen pistemäisten havaintojen ketjua. Liikeratatiedossa pisteistä tunnetaan koordinaattitiedon lisäksi yleensä aikaleima tai vähintään järjestysnumero. Pisteet muodostavat reitin ajan ja tilan läpi, joka voi leikata itseään spatioalisesti.

**Telemetry** tarkoittaa jonkin asian mittaamista ja mittaustiedon lähettämistä tietoliikenneverkon välityksellä esimerkiksi keskitetylle palvelimelle. Tässä työssä telemetrialla viitataan erityisesti ajoneuvoihin asennettujen sensoreiden tuottamaan tietoon ja sen lähettämiseen palvelimelle.

**Digitaalinen tieverkko** tarkoittaa tieverkon vektorimuotoista paikkatietokuvausta. Tässä työssä digitaalisella tieverkolla tarkoitetaan yleisimmin erityisesti tieverkon topologiasta keskilinjakuvausta, jollaista voidaan käyttää esimerkiksi ohjelmalliseen reitinhakuun.

**Tieverkon segmentti** tarkoittaa digitaalisen tieverkon pienintä yksittäistä osaa eli viivaa tai kaarta, joka yhdistää kaksi pistettä.

**Topologia** tarkoittaa paikkatietojärjestelmissä sellaisia kohteiden spatioalisia ominaisuuksia tai kohteiden välisiä spatioalisia suhteita, jotka eivät riipu niiden metrisistä ominaisuuksista tai mittakaavasta. Tällaisia ovat esimerkiksi yhdistyvyys, vierekkäisyys tai päällekkäisyys. Paikkatietokannoissa topologisista ominaisuuksista voidaan johtaa tiedon eheyssääntöjä, kuten vaatimus viivamaisten kohteiden yhdistyvyydelle.

**Karttasovitus** tarkoittaa esimerkiksi ajoneuvon tai muun liikkuvan kohteen liikeratatiedon ohjelmallista sovittamista digitaaliseen tieverkkoon, minkä seurauksena saadaan pääteltyä kohteen kulkema todennäköisin reitti.



# 1 Johdanto

## 1.1 Työn taustat

Vantaan kaupungin kuntatekniikan keskus rakentaa ja ylläpitää katuja ja muita yleisiä alueita. Kaupungin ylläpidossa on noin 1500 kilometriä teitä, katuja ja kevyen liikenteen väyliä. Kaupunki hoitaa itse seitsemästä suuralueesta viittä, ja Hakunilan sekä Korson suuralueet ovat yksityisten alueurakoitsijoiden ylläpidettävänä. Lisäksi kaupungin alueella toimii Liikennevirasto, joka vastaa valtateiden kunnossapidosta sekä Finavia, joka ylläpitää lentoaseman katualueet. Kunnossapitotoimista erityisesti katujen talvikunnossapito vaatii paljon resursseja ja nousee joka vuosi julkiseksi puheenaiheeksi. Julkisen talouden kustannuspaineet aiheuttavat tarpeen tehostaa toimintaa kaikilla osa-alueilla. Yleisten alueiden kunnossapito on merkittävä menoerä kaupungin teknisellä sektorilla, josta Vantaalla vastaa maankäytön, rakentamisen ja ympäristön toimiala. (Vantaan kaupunki, Vättö 2015.)

Yleisten alueiden ylläpito ja erityisesti talvikunnossapito vaatii paljon ajoneuvoja ja erityiskalustoa. Tämän kaluston käytön seuraamiseksi, töiden tehostamiseksi ja kustannusten vähentämiseksi Vantaan kaupunki on hankkinut kilpailutuksen jälkeen vuonna 2014 PPCT Finland Oy:ltä kalustonhallintaratkaisun, joka perustuu telemetriatietojen lähettämiseen kaupungin ajoneuvoilta keskitetylle palvelimelle. Kalustonhallintajärjestelmä perustuu suurelta osin satelliittipaikannukseen, ajoneuvoihin liitettyihin sensoreihin sekä tietoliikenneyhteyteen datan siirtämiseksi. Järjestelmä kattaa kaikki kaupungin ajoneuvot työkoneista kuorma-autoihin ja henkilöautoihin.

Järjestelmään kuuluu internet-selaimella käytettävä sovellus, jossa on reaaliaikainen karttanäkymä koneiden sijainnista, ja mahdollisuus luoda raportteja ajoneuvojen käytöstä, kuten tietyllä aikavälillä kuljettu matka tai arvioitu polttoaineen kulutus. Järjestelmää hyödynnetään esimerkiksi kaupunkiin kohdistuvien talvikunnossapidon tai sen puutteen aiheuttamien vahingonkorvausvaatimusten arviointiin. Valmiin selainsovelluksen lisäksi järjestelmään kuuluu rajapinta, josta on mahdollista hakea tietoja ohjelmallisesti ja analysoida niitä haluumallaan tavalla tai hyödyntää muissa sovelluksissa. Työn motivaationa on tämän datan laajempi hyödyntäminen katujen kunnossapitotöiden seurannan osalta ja tilannetietoisuuden kasvattaminen kaupungissa. Lisäksi kuntalaisille halutaan tarjota läpinäkyvästi tietoa kaupungin toiminnasta visualisoinneilla avoimissa karttapalveluissa.

## 1.2 Työn tavoitteet

Työn ensisijaisena motivaationa on saada rakennettua tietovarasto, josta voidaan hakea telemetriatietoon perustuvaa tilannetietoa Vantaan kaupungin katujen kunnossapidosta. Tämä tarkoittaa käytännössä tietokantaa, joka sisältää tiedon siitä, mitä kunnossapitotoimenpiteitä, kuten auraus, hiekoitus, tai harjaus, on tehty millekin kadulle ja mihin aikaan. Tietokannasta voidaan luoda päivittyvä karttanäkymä, josta ilmenee, mitä ajoneuvokalustolla on viime aikoina tehty, tai onko tietyille katuosuudelle suoritettu kunnossapitotoimenpiteitä. Tietokantaan on myös mahdollista tehdä kyselyitä perustuen kadun nimeen tai spatiaaliseen ja ajalliseen rajaukseen.

Tutkimuksessa analysoidaan paikannusdataa, joka perustuu sellaisiin Vantaan kaupungin ajoneuvoihin, jotka suorittavat talvikunnossapitotöitä kaupungin kaduilla ja kevyen liikenteen väylillä. Erityistä huomiota kiinnitetään auraukseen ja hiekoitukseen, mutta myös muun tilatiedon tapahtumia voidaan seurata ja analysoida. Tutkimuksessa käytettävä data on peräisin 15 kuorma-autosta ja 9 työkoneesta ja 13 henkilöautosta. Työkoneilla tarkoitetaan tässä tutkimuksessa eri kokoisia traktoreita ja pyöräkuormaajia. Näiden kahden ajoneuvoryhmän ajokäyttäytymisessä on huomattavia eroja, mitkä johtuvat siitä, että kuorma-autoja käytetään katuverkon kunnossapidossa, mutta työkoneita käytetään sekä katu- että kevyen liikenteen verkon kunnossapidossa. Liitteessä 1 on esitetty kuvat tavanomaisesta talvikunnossapidossa käytettävästä kuorma-autosta ja pyöräkuormaajasta.

Jotta kuvattu tietokanta ja kartta on mahdollista luoda, on luotava järjestelmä, joka kerää ajoneuvojen analysoitavaa telemetriadataa ja esikäsittelee sen sellaiseen muotoon, joka voidaan visualisoida ja johon voidaan kohdistaa tehokkaasti kyselyitä. Satelliittipaikannukseen perustuvan telemetriadatan synnyttämän havaintopistejoukon muokkaaminen analysoitavaan muotoon edellyttää käytännössä *karttasovitukseksi* kutsutun ohjelmallisen käsittelyn tekemistä. Karttasovituksessa paikannusdatasta muodostuva liikerata ja digitaalinen tieverkko sovitetaan yhteen siten, että saadaan selville todellinen reitti, jonka ajoneuvo on kulkenut. Edellä mainittuun yleiseen tavoitteeseen liittyen yksi työn päätavoitteista on sopivan karttasovitusalgoritmin löytäminen ja toteuttaminen kaupungin paikkatietoinfrastruktuuriin.

Työn tutkimuskysymykset ovat:

- Millaisia algoritmisia ratkaisuja voidaan käyttää karttasovituksen toteuttamiseksi?
- Millaista Vantaan kaupungin telemetriajärjestelmän tuottama data on?
- Onko luotettava karttasovitus mahdollista tehdä sillä datalla, mitä Vantaan kaupungin telemetriajärjestelmä tuottaa?
- Minkälaista epätarkkuutta karttasovitetussa datassa esiintyy verrattuna todellisuuteen ja miten sitä kannattaa mitata?

### **1.3 Tutkimusmenetelmät**

Työssä sovelletaan konstruktiivista tutkimusmenetelmää, jossa pyritään ratkaisemaan katu- ja kunnossapidon tilannetiedon kannalta keskeinen karttasovitusongelma mahdollisimman laadukkaasti sekä mittaamaan sen onnistumista. Tutkimusmetodina on kirjallisuuskatsaus, jonka pohjalta pyritään tekemään perustellut valinnat käytettävien tekniikoiden osalta. Empiirisessä osiossa toteutetaan tiedon keruu ja lupaavimmat karttasovitusalgoritmit sekä tutkitaan niiden sovitustarkkuutta kvantitatiivisesti verrattuna manuaalisesti kerättyyn ja luokiteltuun dataan.

### **1.4 Työn rakenne**

Työn alkupuolen luvuissa 2 ja 3 esitellään liikennetelematiikkaa ja paikannusmenetelmät, jotka ovat usein liikennetelematiikan sovellusten keskeisiä teknologioita. Luvussa 4 esitellään karttasovitusongelma ja tehdään kirjallisuuskatsaus sen selvittämiseksi, miten ongelma voidaan ratkaista sekä mitkä ovat eri menetelmien heikkoudet ja vahvuudet ja minkä tyyppiin ongelmiin niitä voidaan soveltaa. Luvussa 4 kerättyjen tietojen perusteella päätetään

toteutettavat menetelmät, joiden teoriaa esitellään luvussa 5. Luvut 6 ja 7 ovat työn konstruktiivista tutkimusosaa, joissa perehdytään ensin tutkittavaan dataan ja sen jälkeen esitellään luodut karttasovitusmenetelmät sekä toteutukseen käytetyt teknologiat. Lisäksi luvussa 7 vertaillaan algoritmien sovitustarkkuutta ja suorituskykyä. Luvussa 8 kuvaillaan karttasovituksen hyödyntämistä integroimalla valittu algoritmi operatiiviseen järjestelmään jatkuvan ja ajantasaisen tilannetiedon sisältävän tietovaraston luomiseksi. Luvuissa 9 ja 10 tehdään yhteenveto tuloksista ja esitetään kehitysehdotukset järjestelmän parantamiseksi sekä pohdinnat jatkotutkimuksista.

## 2 Liikennetelematiikka

Telematiikalla tarkoitetaan tiedon siirtoa tietoliikenneverkon kautta ja tiedon laskennallista prosessointia. Sana tulee ranskan *télématique*-sanasta, jossa yhdistyy sanat *télécommunication* ja *informatique*. Telemetria-sanaa käytetään sekä arkikielessä että kirjallisuudessa yleisesti synonyymina telematiikalle. Toisaalta telemetria tarkoittaa erityisesti jonkin asian mittaamista ja mittaustiedon siirtämistä tietoliikenneverkon kautta (esimerkiksi analysoitavaksi keskitetylle palvelimelle). (Goel 2008.)

Tietotekniikan kehityksen ansiosta tietokoneista on tullut 2000-luvulla niin pieniä, tehokkaita ja energiatehokkaita, että niitä on mahdollista liittää lähes mihin tahansa laitteeseen tai paikkaan. Langattomien tietoliikenneyhteyksien kehittyminen mahdollistaa entistä suurempien datamäärien siirron reaaliaikaisesti liikkuvilta laitteilta keskitetyille palvelimille. Myös ajoneuvoihin liitettävien sensoreiden suorittama datan keräys ja sen prosessointi sekä siirto langattomasti keskitetyille palvelimille on mahdollista näiden kehityssuuntien ansiosta. (Goel 2008.)

Telematiikan hyödyntämistä liikenteeseen tai ajoneuvoihin liittyvään organisointiin, tiedonkeruuseen tai kontrollointiin kutsutaan liikennetelematiikaksi. Toisaalta koko telematiikka-sana on nykyisin vakiintunut tarkoittamaan usein nimenomaan ajoneuvoihin ja liikenteeseen liittyvää telematiikkaa. Älyliikennettä (engl. Intelligent Transportation Systems, ITS) käytetään toisinaan synonyymina liikennetelematiikalle, mutta älyliikennettä voidaan pitää laajempänä aihealueena liittyen tietotekniikan hyödyntämiseen liikenteessä. Liikennetelematiikka kuuluu osaksi älyliikennettä. (Goel 2008.)

Ajoneuvoja, jotka kytkeytyvät toisiinsa ja liikenteeseen liittyviin järjestelmiin ja palveluihin internetin välityksellä, voidaan pitää esineiden internetin (Internet of Things, IoT) osana. IoT tarkoittaa sensoreiden, tuotantolaitteiden, kodinelektroniikan ja muiden vastaavien internetiin kytkeytyneiden esineiden muodostamaa verkkoa. Mitchell et al. (2010) kutsuu ajoneuvoista ja liikennetelematiikan järjestelmistä muodostuvia verkkoja liikenteen internetiksi (Mobility Internet), joka tulee mahdollistamaan ajoneuvojen välisen reaaliaikaisen, georeferoidun tiedonvaihdon ja muuttamaan liikenteen paradigmaa merkittävästi lähivuosina. Tietoliikenneteknologia on myös yksi autoteollisuuden vahvoista tulevaisuuden suuntauksista. Teknologian tutkimusyritys Gartner ennustaa, että internet-yhteyksin varustettuja ajoneuvoja on vuoteen 2020 mennessä yli 250 miljoonaa tehden niistä yhden IoT:n pääelementeistä (Gartner 2015).

### 2.1 Sovellukset

Yleisimpiä liikennetelematiikan sovelluksia ovat erilaiset liikenne- ja matkustustietosovellukset kuten paikkahaku, käyttäjän paikannus ja reititys ajoneuvon navigaattorilla. Tyypillisesti sovelluksissa hyödynnetään satelliittipaikannusta ja digitaalista karttaa, jotta käyttäjälle voidaan tarjota reaaliaikaisia ajo-ohjeita auditiivisesti ja visuaalisesti. Navigointisovelluksissa voidaan lisäksi hyödyntää esimerkiksi reaaliaikaisia ruuhka-, onnettomuus- tai tietyötietoja, mikäli älykkään liikenteen infrastruktuuri ja sen palvelut, sekä käytettävä laitteisto tukee tällaista. (Goel 2008.)

Kaupallisiin kuljetussovelluksiin liittyviä liikennetelematiikkasovelluksia on alettu hyödyntää kilpailun rajoitusten poistamisen seurauksena, kun erityisesti vanhat logistiikka-alan toimijat ovat globalisaatiokehitykseen vastataksaan pyrkineet saavuttamaan kilpailuetuja uusiin kilpailijoihin nähden. Tällaisiin sovelluksiin kuuluvat erilaiset kalustonhallintaan (fleet management) ja reittien tai töiden suunnitteluun liittyvät toiminnot. Kalustonhallinnan sovelluksiin kuuluu esimerkiksi kaluston käyttöasteen seuranta ja kaluston kunnon seuranta (Said et al. 2014). Kilpailuetuja voidaan saavuttaa muun muassa tarjoamalla niin kutsuttuja juuri-oikeaan-aikaan -palveluita (Just-In-Time, JIT) mahdollistaen inventaarioiden ja varastojen pienentämisen sekä kaluston käyttöasteen nostamisen. Tällaiset sovellukset edellyttävät reaaliaikaista seuranta, paikannusta ja reititystä koko ajoneuvokalustolle. (Goel 2008.)

Etädiagnostiikkasovelluksilla voidaan tarkkailla ajoneuvon tilaa etänä. Tällaisten sovellusten käyttötarkoituksena on muun muassa ajoneuvojen vioittumisen ennaltaehkäisy. Ajoneuvo voidaan kutsua varikolle tai huoltoon, mikäli sen lähettämässä diagnostiikkatiedossa havaitaan poikkeamia. Tällainen poikkeama voi olla esimerkiksi ajoneuvon eri nesteiden määrään tai kulumisnopeuteen liittyvä tieto. Etädiagnostiikka toimii joko asentamalla erillisiä sensoreita ajoneuvoon tai kytkemällä telemetriajärjestelmä ajoneuvon CAN-väylään (Controller Area Network), jolloin voidaan lukea ja lähettää ajoneuvon ajotietokoneen tuottamaa dataa, esimerkiksi OBD-diagnoositietoja (On-Board Diagnostics). OBD-järjestelmästä on mahdollista saada tietoa esimerkiksi moottorin kierrosluvuista, lämpötilasta ja vikakoodeista. Kokonaisessa kalustonhallintaratkaisussa tällaisilla toimenpiteillä voidaan säästää merkittävästi huoltokustannuksissa, mikäli ajoneuvojen rikkoutumista saadaan vähennettyä ja huoltoja ajoitettua oikea-aikaisemmiksi. (Cassias ja Kun 2007, Said et al. 2014.)

Etädiagnostiikan kanssa samaan kategoriaan voidaan asettaa erilaisista sensoreista kerättävät kaupalliseen toimintaan liittyvät tilatiedot ajoneuvosta, jotka liittyvät esimerkiksi tiettyyn työtehtävään. Katujen kunnossapidon näkökulmasta tällaisia ovat esimerkiksi ajoneuvoon kytkettyjen lisälaitteiden ja niiden asentotietojen perusteella päätellyt työtehtävät, kuten avaraus, harjaus, kadunpesu tai hiekoitus.

Turvallisuuteen liittyviä sovelluksia ovat esimerkiksi varoitukset onnettomuuksista, eläimistä tai moottoritiellä poikkeuksellisen hitaasti tai väärään suuntaan liikkuvista ajoneuvoista. Yksi turvallisuuteen liittyvä sovellus on automaattinen hälytysjärjestelmä ja koordinaattien siirto hätäkeskukseen onnettomuustilanteessa, esimerkiksi turvatyynyn lauetessa tai muiden törmäykseen liittyvien järjestelmien aktivoituessa (Goel 2008). Hälytysajoneuvoissa ja liikennevaloissa voidaan hyödyntää järjestelmää, joka pyrkii varmistamaan hälytysajoneuvoille esteettömän reitin onnettomuuspaikalle estämällä muun liikenteen kulkemisen liikennevaloilla ohjatuissa risteyksissä tai rampeilla. Xun (2000) mukaan tämänkaltaiset järjestelmät voivat nopeuttaa pelastustoimia jopa 30 %. Oulun kaupunki toteutti vuonna 2015 HALI-järjestelmän, joka hyödyntää hälytysajoneuvojen keskitetylle palvelimelle lähettämää satelliittipaikannustietoa ja takaa niille vihreän valon jokaiseen liikennevalo-ohjattuun risteykseen ajoneuvon oletetulle reitille (ESNC 2015).

Liikennemaksut ovat eräs Suomessakin keskustelua herättänyt liikennetelematiikan sovellus. Niiden ajatuksena on pääasiassa vähentää henkilöautoliikenteestä koituvia haittoja (etenkin ruuhkautumista tai saasteita) alueellisesti asettamalla tietyille tieosuuksille tai alueille kulkemiselle ylimääräisiä kustannuksia. Helsingin Seudun Liikenteen (HSL) selvityksen mukaan maksuja voisi kerätä joko sähköisellä vinjetillä eli aikaperusteisella ajoluvalla tietyllä alueella, portteihin tai maksupisteisiin perustuvalla järjestelmällä, jossa ajoneuvot

tunnistetaan joko mikroaaltotekniikalla tai rekisterikilpien kuvantunnistuksella, tai satelliittipaikannukseen ja kilometrimaksuihin perustuvalla tekniikalla (HSL 2016). Ollilan et al. (2013) työryhmän selvityksen mukaan Suomessa kannattaisi siirtyä satelliittipaikannukseen perustuviin ruuhkamaksuihin ja ajoneuvoverotukseen. Satelliittipaikannukseen perustuviin kilometrimaksuihin siirtyminen antaisi valtiolle tehokkaan liikennepoliittisen välineen ja auttaisi kohdistamaan liikenteeseen liittyvät verot ja maksut juuri siten kuin on tarpeellista ja oikeudenmukaista.

Tulevaisuudessa autonomisesti liikkuvien ajoneuvojen järjestelmät nojaavat vahvasti liikennetelematiikan sovelluksiin. Ajoneuvon paikannus tieverkkoon ja reititys ovat ajoneuvon kamera-, tutka- ja lasersensoreiden ohella olennaisia autonomisen ajon mahdollistavia teknologioita. Tieverkon ja sen olosuhteiden informaation välitys ajoneuvoille, mutta myös autonomisten ajoneuvojen keskenään ja tavallisten autojen kanssa vaihtama informaatio voi tulevaisuudessa olla olennainen osa autonomista liikennöintiä (Bazzan ja Klügl 2013). Myös tavallisten ajoneuvojen erilaiset yleistyvät avustusjärjestelmät, kuten kaistavahdit ja automaattiset hätäjarrutusjärjestelmät, voivat nojautua samoihin teknologioihin.

Liikennetelematiikan järjestelmien tuottaman datan saatavuuden kasvun myötä sitä käytetään nykyään myös yhä enemmän tiedonlouhintaan. Tiedonlouhinta tarkoittaa ennalta tuntemattomien kuvioden tai riippuvuussuhteiden löytämistä datasta sellaisia automatisoituja menetelmiä hyödyntäen, jotka eivät ole riippuvaisia *a priori* tiedosta (Miller ja Han 2001). Tätä edesauttaa myös kulttuuri, jossa julkinen hallinto julkaisee avoimena datana omien tietokantojensa sisältöä. Suomessa tällaista dataa on saatavilla esimerkiksi pääkaupunkiseudulta Helsinki Region Infoshare portaalista (hri.fi). Liikennetelematiikan sovellusten keräämästä paikannusdatasta voidaan louhia erityisesti liikennejärjestelmän toimivuuteen ja ihmisten käyttäytymiseen liittyvää tietoa, jota voidaan käyttää esimerkiksi yhdyskuntasuunnittelun tai palveluiden kehittämisen lähtötietona. Esimerkiksi Zheng et al. (2011) analysoivat Pekingin taksien paikannusdataa ja pyrkivät tunnistamaan alueita, joissa liikennesuunnittelu oli tavalla tai toisella epäonnistunut, jotta nämä epäkohdat voisi ottaa huomioon tulevissa hankkeissa. Toinen sovellus, jossa tieliikenteestä kerättyä paikannusdataa voidaan hyödyntää, on liikenneverkon ruuhkamallinnus, jota nykyään tehdään yleensä niin sanotulla aktiivisen kelluvan auton menetelmällä, eli ajamalla liikennevirran mukana yksittäisellä ajoneuvolla, joka kerää dataa liikenteen sujuvuudesta (Kytö 2015). Tällainen työläs matka-aikatiedon manuaalinen kerääminen voidaan korvata käyttämällä dataa tarpeeksi laadukkaista ja kattavista paikkatietokannoista.

Karttasovitus liittyy olennaisena osana moniin liikennetelematiikan sovelluksiin, erityisesti kuljettajaa avustaviin navigointisovelluksiin sekä kalustonhallintaan ja turvallisuuteen liittyviin etädiagnostiikkaa hyödyntäviin sovelluksiin. Sitä käytetään ajonaikaisissa sovelluksissa paikannustarkkuuden parantamiseen ja toisaalta erilaisissa jälkilaskentana suoritettavissa datan analysointisovelluksissa virheiden poistamiseen paikannusdatasta ja tiedon muokkaamiseen sellaiseen muotoon, jossa sitä voidaan jatkossa hyödyntää (Kim et al. 2000, Xu et al. 2014). Tämän tutkimuksen sovellusalue kuuluu jälkimmäiseen luokkaan ja sen päätarkoituksena on ajoneuvojen sensoritietojen etälukeminen ja sijaintitiedon analysointi.

## 2.2 Yksityisyydensuoja

Käyttäjien ja kansalaisten yleinen asenne telematiikkaa hyödyntäviä sovelluksia kohtaan riippuu hyvin pitkälti siitä, miten he pystyvät luottamaan palveluntarjoajaan yksityisyys- ja tietoturva-asioissa. Käyttäjien pitää vakuuttua siitä, ettei järjestelmän tai palvelun tarjoaja käytä tietoja vihamielisiin tarkoituksiin. Lisäksi palveluntarjoajan tulee varmistua siitä, etteivät laitteet tai tietoliikenneyhteydet ole haavoittuvaisia tietoturvauhille (Cassias ja Kun 2007). Innamaa et al. (2015) tekemän kyselyn perusteella 43 % haastatelluista suomalaisista oli huolissaan satelliittipaikannukseen perustuvan liikennemaksujärjestelmän tietoturvasta.

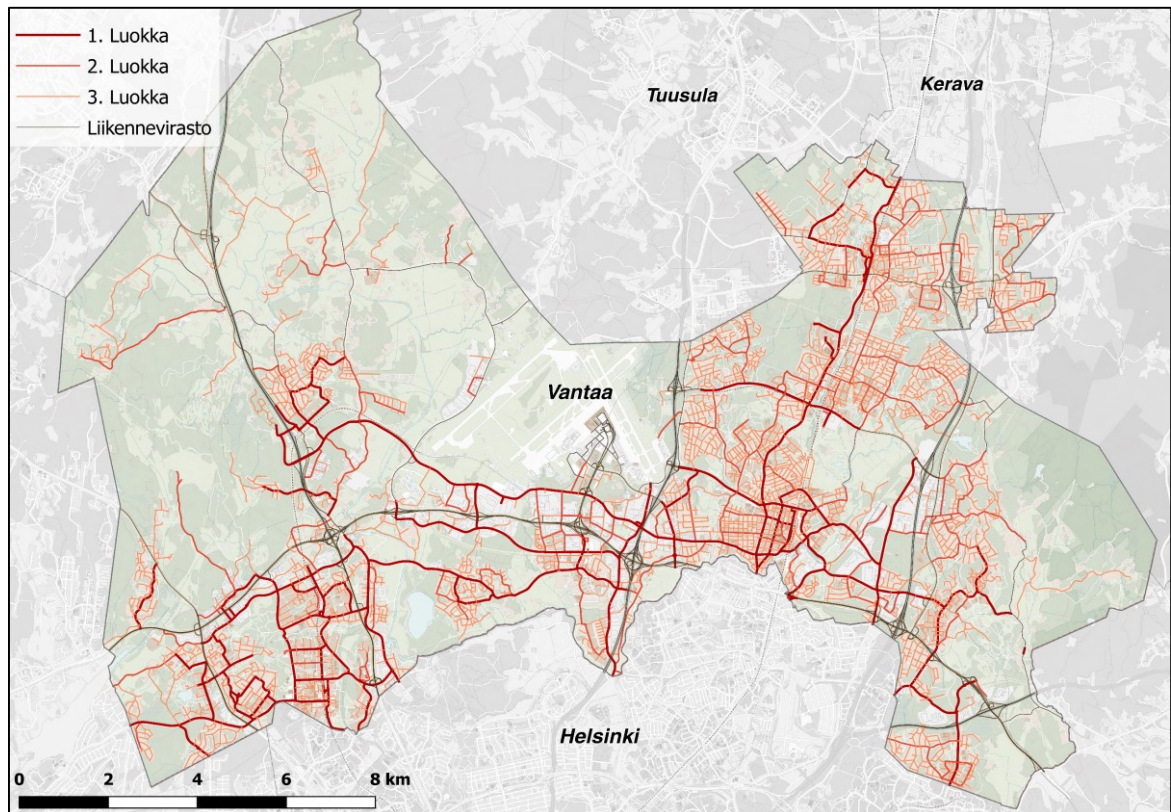
Laukkanen (2015) tutki diplomityössään Helsingin kaupungin rakennuspalveluiden Staran telematiikkajärjestelmän käyttöönottoa käyttäjien näkökulmasta. Suurin yksittäinen teknologian vastustukseen liittyvä seikka oli huoli käyttäjien yksityisyydestä. Erityisesti kokonaisten kuljettujen reittien tallennusta pidettiin arveluttavana "kyttäysteknologiana". Tämä johtuu ensisijaisesti siitä, että kuljetusta reitistä on mahdollista jälkikäteen tutkia yksityiskohtaisesti yksittäisen henkilön toimia päivän aikana. Järjestelmän vastustajat mielsivät sen ensisijaisesti henkilönseurantajärjestelmäksi kalustonseurannan sijaan (Laukkanen 2015). Myös VTT:n tietoturvaraportissa (Lehtonen et al. 2016) ehdotetaan muun muassa erottamaan käyttäjien identiteetti ja paikkatiedot toisistaan, mikäli eri osajärjestelmät eivät tarvitse molempia tietoja.

Työelämän tietosuojalain 21 §:n mukaan työntekijöihin kohdistuvan kameravalvonnan, kulunvalvonnan ja muun teknisin menetelmin toteutetun valvonnan tarkoitus, käyttöönotto ja valvonnassa käytettävät menetelmät kuuluvat yhteistoimintamenettelyn piiriin (L 759/2004). Vantaan kaupunki loi vuoden 2012 lopussa yhteistoimintalain mukaisesti yhdessä henkilöstön edustajien kanssa koko kaupungin henkilöstöä koskevat käytöissännöt ajoneuvojen seurantajärjestelmälle ennen järjestelmän käyttöönottoa. Käytöissännöissä todetaan, että kyse on välillisestä paikantamisesta, jossa paikannustiedot ja henkilötiedot eivät linkity suoraan toisiinsa, vaikkakin se olisi mahdollista tehdä yhdistelemällä eri tietolähteitä, kuten työvuorolistoja aineistoon. Henkilöstön edustajille tärkeää oli erityisesti se, ettei paikannustietoja käytetä yksittäisen henkilön työajan seurantaan, ja tämä kohta kirjattiin käytöissännöihin. Käytöissännöjen mukaan järjestelmän käyttötarkoituksena on muun muassa tehostaa ja helpottaa työn ohjausta, todentaa työsuorituksia, tarkentaa raportointia, parantaa kustannustehokkuutta ja resurssien käyttöä, seurata polttoaineen kulutusta ja kaluston käytöstä, sekä turvata kaupungin omaisuutta. (Vantaan kaupunki 2012.)

Kaikissa telematiikkaa, paikannusta ja erityisesti kokonaisten reittien mittaamista ja tallentamista koskevissa sovelluksissa onkin syytä ottaa tarkasti huomioon yksityisyydensuojaan liittyvät seikat sekä lain että järjestelmän käyttäjien motivaation ja teknologian omaksumisen kannalta. Tämän tutkimuksen aiheena olevassa sovelluksessa ei kerätä lainkaan henkilötietoa, vaan mielenkiinto kohdistuu katuihin ja niille tehtyihin kunnossapitotoimenpiteisiin. Järjestelmästä on mahdollista selvittää, mikä ajoneuvo on liikkunut tiettyssä paikassa tiettyyn aikaan, jonka tiedon yhdistämällä muuhun tietoon on mahdollista saada selville myös yksittäisen henkilön liikkeitä. Henkilötietoa ei kuitenkaan liitetä järjestelmään, eikä sen edes tarvitse lähtökohtaisesti välittää yksilöityjen ajoneuvojen tietoa, vaan huomio kohdistuu katuverkkoon ja sen kunnossapitotilanteeseen. Paikannustiedon keräämiseen liittyvän arkaluontoisuuden vuoksi selkeiden pelisääntöjen luominen ja henkilöstön informointi tiedon keräämisen motivaatiosta ja käytöstä ovat kuitenkin joka tapauksessa suositeltavia toimenpiteitä.

### 2.3 Kehitettävän sovelluksen erityishaasteet

Tutkimukseen liittyy muutamia aiheena olevasta sovelluksesta seuraavia erityishaasteita, jotka tekevät ongelmasta erilaisen kuin tavanomaiseen tieliikenteeseen liittyvät karttasovitusmenetelmät. Katujen kunnossapitoa suorittavien ajoneuvojen ajokäyttäytyminen ja niiden käyttämät reitit voivat poiketa tavallisesta henkilö- tai tavaraliikenteestä. Ajoneuvojen liikkeet eivät ole yhtä hyvin ennalta-arvattavissa kuin normaalin tieliikenteen. Tavallisesti liikenneverkkoa käytetään siirtymiseen paikasta toiseen, jolloin on loogista tehdä oletuksia lyhyimpiin tai nopeimpiin reitteihin perustuen, jos matkan määränpää tunnetaan. Katujen kunnossapitoa suorittavien ajoneuvojen tarkoituksena puolestaan on liikkua katuverkossa kattavasti suorittaen samalla jotakin työtehtävää. Esimerkiksi lumisateen jälkeen lähtökohtaisesti kaikilta väyliltä poistetaan lumi ja tarvittaessa kuljetetaan se lumenvastaanottoaikaan. Lumienpoiston järjestys riippuu puolestaan katujen kunnossapitoluokituksista. Kunnossapitoluokitus määrää katujen hoidon tärkeysjärjestyksen, joka perustuu liikennemääriin, joukko liikenteen reitteihin ja liikenteen erityistarpeisiin, kuten terminaaleihin. Kunnossapitoluokkia on kolme, joista 1. luokkaan kuuluvat pääkadut ja tärkeimmät kevyen liikenteen väylät. 2. luokkaan kuuluvat loput linja-autoreitit ja tärkeimmät kokoojakadut ja 3. luokkaan kuuluvat kaikki loput kadut (Vantaan kaupunki). Kuvassa 1 on esitetty Vantaan katuverkko kunnossapitoluokan mukaan teemoitettuna.



Kuva 1: Vantaan kadut kunnossapitoluokan mukaan teemoitettuna.

Yksi erityinen haaste sovelluksessa on ajoneuvojen osittain erittäin tiheässä tieverkossa tapahtuva liikkuminen, mikä johtuu siitä, että kaupungin työkoneet liikkuvat kunnossapito-



töissä myös kevyen liikenteen väylillä, mikä pahimmillaan kaksinkertaistaa tieverkon tiheyden. Nämä reitit ovat myös yleensä rinnakkaisia ja yhdensuuntaisia katuverkon segmenttien kanssa, mikä tekee karttasovituksen vaikeaksi.

Katuverkon kunnossapitoa suorittavien ajoneuvojen nopeudet ovat usein muuta tieliikennettä hitaampia. Tämä vähentää esimerkiksi väylien keskinopeuksien tai nopeusrajoitustietojen tuntemisen arvoa, sillä sovelluksen ajoneuvojen nopeudet eivät välttämättä noudata väylän oletettuja nopeuksia. Erityisesti auraamiseen liittyy ajoittain lisäksi peruuttelua ja paikallaan pyörimistä, jotta koko ajorata ja esimerkiksi bussipysäkkien levennykset saadaan puhdistettua lumesta. (Tiittanen 2016.)

Kunnossapitoa suorittavat ajoneuvot saattavat myös rikkoa liikennesääntöjä töissään. Esimerkiksi siirtyminen yhdeltä hoidettavalta väylältä toiselle saattaa olla nopeinta tehdä ajamalla väärään suuntaan yksisuuntaista katua tai keskikorokkeella erotettua tieosuutta tai esimerkiksi oikaisemalla liikenneympyrän läpi kulkemalla sitä myötäpäivään. Etenkin silloin kun katuverkkoa hoidetaan sellaiseen aikaan, kun muuta liikennettä on hyvin vähän (esimerkiksi pääkatuja yöllä tai asuntokatuja keskipäivällä) tällaiset siirtymät voivat olla hyvinkin mahdollisia. (Tiittanen 2016.)

Nämä erityispiirteet tulee ottaa huomioon karttasovitusta suunniteltaessa, eivätkä kaikki perinteisen tieliikenteen karttasovituksessa käytetyt oletukset välttämättä pidä paikkaansa, kun sovelluskohteena on katujen kunnossapito. Esimerkiksi tieverkoilla väärään suuntaan liikkuminen on huomioitava vähintään siten, ettei koko reitin sovitus epäonnistu tällaisen seurauksena. On myös tiedostettava, ettei ajoneuvojen nopeuteen tai liikenneverkon nopeusrajoituksiin liittyviä tietoja välttämättä voida käyttää hyödyksi samalla tavalla kuin tavallista tieliikennettä koskevissa sovelluksissa, eikä lähtöoletukseksi voida ottaa esimerkiksi mahdollisimman suoria reittivalintoja.

### 3 Paikannus

Ajoneuvon reaaliaikainen sijainnin määrittäminen on yksi liikennetelematiikkasovellusten keskeisimmistä teknologioista. Tässä luvussa käydään läpi menetelmät, joilla ajoneuvojen paikannus yleisimmin tehdään. Sijaintitarkkuudeltaan karkeita tai alueelliselta kattavuudeltaan puutteellisia paikannusratkaisuja, kuten GSM- tai WiFi-tukiasemapaikannus ei käsitellä tässä tutkimuksessa.

#### 3.1 Satelliittipaikannus

Ajoneuvojen sijainnin määrittäminen liikennetelematiikan sovelluksissa perustuu nykyään lähes aina satelliittipaikannukseen. Satelliittipaikannuksella tarkoitetaan sijainnin määrittämistä jonkin satelliittipaikannusjärjestelmän satelliitteja kiintopisteinä käyttäen. Vuonna 2016 operatiivisena on kaksi globaalia satelliittipaikannusjärjestelmää, Yhdysvaltain puolustusministeriön NAVSTAR GPS (Navigation System with Time and Ranging Global Positioning System, yleensä vain GPS) johon kuuluu normaalitilanteessa 24 satelliittia, sekä Venäjän GLONASS (Globalnaya navigatsionnaya sputnikovaya sistema), johon kuuluu 24 satelliittia. Muita kehitteillä olevia globaaleja järjestelmiä, jotka eivät kuitenkaan vielä ole täysin operatiivisia, ovat Euroopan avaruusjärjestön Galileo ja Kiinan Compass. Molempien järjestelmien arvioidaan kuitenkin olevan operatiivisia vuoteen 2020 mennessä (Geodeettinen laitos). Yhteisnimitys eri satelliittipaikannusjärjestelmille on GNSS (Global Navigation Satellite System). (Poutanen 2016.)

Tässä luvussa esitellään GPS-järjestelmään perustuva sijainnin määrittäminen. Vaikka muissa järjestelmissä tekniikat ovat hieman erilaisia, tapahtuu paikannus pääpiirteissään samalla tavalla kaikissa satelliittipaikannusjärjestelmissä. GPS on vanhin satelliittipaikannusjärjestelmä ja se on ollut täysin operatiivinen vuodesta 1995 alkaen. Järjestelmässä on nykyään 24 satelliittia ja 6–8 varasatelliittia, jotka kiertävät maapalloa noin 20 000 km korkeudella kuudella eri ratatasolla, jotka ovat noin 55 asteen inkliinaatiokulmassa päiväntasaajaan nähden. Näin saavutetaan täydellinen kattavuus ympäri maapallon. Paikannusta varten vähintään neljän satelliitin on oltava näkyvissä vastaanottimelta katsottuna. (Poutanen 2016.)

Satelliittipaikannus perustuu pohjimmiltaan ajanmittaukseen, jonka johdosta satelliiteissa käytetään tarkkoja atomikelloja, joita jokaisessa satelliitissa on 2–4 (Seeber 2003, s. 304–307). Satelliitit ovat yhteydessä järjestelmän maasegmenttiin kuuluviin kontrollikeskuksiin, jotka tarkkailevat satelliittien tilaa ja määrittelevät satelliittien kellovirheen ja lähetettävät ratatiedot. Yksittäinen GPS-satelliitti lähettää signaalia 2–3:lla eri taajuustasolla, L1 (1575,42 MHz) ja L2 (1227,6 MHz) sekä uudemmissa satelliiteissa L5 (1176,45 MHz). Signaaleihin on moduloitu koodit, joita tarvitaan GPS-vastaanottimen paikantamiseen. L1-signaali sisältää C/A-koodin (Coarse Acquisition), joka on tarkoitettu siviilikäyttöön. Lisäksi molemmat signaalit sisältävät salatun P-koodin, joka on tarkoitettu Yhdysvaltain sotilaskäyttöön. Uudemmissa satelliiteissa C/A-koodi lähetetään sekä L1- että L2-taajuuksilla ja sotilaskäyttöön on uusi M-koodi. (Poutanen 2016, Seeber 2003, s. 218–222.)

Aina 2000-luvulle asti C/A-koodissa oli käytössä signaalin tahallinen heikennys (Selective Availability, SA), joka kasvatti paikannuksen epätarkkuuden noin viisinkertaiseksi, eli muutamia kymmeniä metreihin. Vuonna 2000 tahalliset heikennykset kuitenkin poistettiin, minkä jälkeen paikannustarkkuus C/A-koodilla on 13 metriä (95 % luottamusväli). (Poutanen 2016, Yhdysvaltain puolustusministeriö 2008.)

Satelliittien lähettämiin signaaleihin on moduloitu myös navigaatioviesti, josta on luettavissa yleistä tietoa satelliitin ja koko järjestelmän tilasta, satelliitin ratatiedot sekä karkeat tiedot muiden satelliittien radoista ja sijainneista (niin sanottu almanakka). (Poutanen 2016, s. 170.)

### 3.1.1 Sijainnin määrittäminen

Tavanomaisissa siviilikäyttöön tarkoitetuissa yhdellä vastaanottimella tapahtuvassa paikannuksessa paikannus perustuu C/A-koodiin. Tätä kutsutaan absoluuttiseksi paikannukseksi tai koodipaikannukseksi. Paikannuksessa pyritään ratkaisemaan vastaanottimen kolmiulotteiset koordinaatit  $x$ ,  $y$  ja  $z$ . Paikannus tehdään määrittämällä satelliittien ja vastaanottimen väliset etäisyydet, mikä taas perustuu aikaerojen mittaamiseen. Tästä johtuen esimerkiksi sekunnin ero vastaanottimen ja satelliitin kellojen ajan välillä aiheuttaa noin 300 000 km eron etäisyyden laskentaan, koska satelliitin lähettämä radiosignaali kulkee likimain valon nopeutta (noin 0,3 m yhdessä nanosekunnissa eli  $10^{-9}$  sekunnissa). Vastaanotin pyrkii yleensä synkronoimaan kellonsa GPS-satelliittien aikaan, mutta pienikin kellovirhe aiheuttaa suuria paikannusvirheitä. Kellovirheitä lisäksi syntyy käytännössä aina tavanomaisten vastaanotinlaitteiden heikkolaatuisten kellojen johdosta. Tämä ongelma ratkeaa siten, että vastaanottimen kellovirhe otetaan mukaan laskentaan neljäntenä tuntemattomana muuttujana ratkaistavien koordinaattien lisäksi. Neljännen tuntemattoman muuttujan lisääminen yhtälöryhmään aiheuttaa myös vaatimuksen neljän satelliitin signaalin yhtäaikaistamiselle. Aikaeron perusteella laskettuja etäisyyksiä satelliitteihin kutsutaan pseudoetäisyyksiksi. Saadaan yhtälöryhmä:

$$\begin{cases} P1 = \sqrt{(x_1 - X)^2 + (y_1 - Y)^2 + (z_1 - Z)^2} + cd_t \\ P2 = \sqrt{(x_2 - X)^2 + (y_2 - Y)^2 + (z_2 - Z)^2} + cd_t \\ P3 = \sqrt{(x_3 - X)^2 + (y_3 - Y)^2 + (z_3 - Z)^2} + cd_t \\ P4 = \sqrt{(x_4 - X)^2 + (y_4 - Y)^2 + (z_4 - Z)^2} + cd_t \end{cases} \quad (1)$$

missä  $X$ ,  $Y$ ,  $Z$  ovat vastaanottimen tuntemattomat koordinaatit,  
 $P1$ ,  $P2$ ,  $P3$ ,  $P4$  ovat pseudoetäisyydet satelliitteihin,  
 $d_t$  on vastaanottimen kellovirhe,  
 $c$  on signaalinopeus ja  
 $x_i$ ,  $y_i$ ,  $z_i$  ovat C/A-koodista luetut satelliittien sijainnit.

Yhtälöryhmä linearisoidaan ja ratkaistaan iteratiivisesti pienimmän neliösumman menetelmällä käyttäen aina edellistä paikkaratkaisua alkuarvauksena uudelle kierrokselle (Seeber 2003, s. 136).

### 3.1.2 Tarkkuus ja virhelähteet

Tuntemattomien muuttujien estimoinnissa tapahtuu virheitä, jos mallissa ei pystytä ottamaan huomioon kaikkia fysikaalisia ilmiöitä, jotka vaikuttavat satelliitin ja vastaanottimen välisen etäisyyden määrittämiseen. Näitä virheitä kutsutaan nimellä UERE (User Equivalent Range Error). Virhelähteitä ja niiden aiheuttamien paikannusvirheiden kokoluokkia on eritelty taulukossa 1.

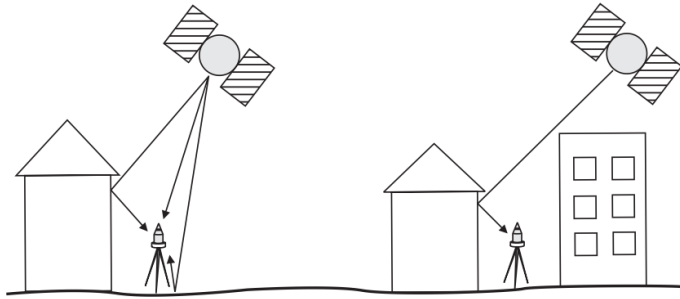
| <i>Avaruudessa vaikuttavat virheet</i>      | <i>Virheen kokoluokka</i> |
|---|---------------------------|
| Satelliittien kellovirheet                  | 1-2 m                     |
| Satelliittien ratavirheet                   | 1-2 m                     |
| <i>Ilmakehässä vaikuttavat virheet</i>      |                           |
| Ionosfäärin vaikutukset                     | 1-10 m                    |
| Troposfäärin vaikutukset                    | < 1 m                     |
| Monitieheijastumat                          | 1-5 m                     |
| <i>Vastaanottimella vaikuttavat virheet</i> |                           |
| Havainnoinnin ja laskennan virheet          | 1-5 m                     |
| Antennin vaihekeskipisteen virhe            | < 10 cm                   |

Taulukko 1: GPS-paikannuksen virhelähteet (Seeber 2003, Yhdysvaltain puolustusministeriö 2008).

GPS-järjestelmän kontrollointikeskukset huolehtivat avaruudessa vaikuttavien virheiden korjaamisesta. Näihin korjauksiin kuuluu esimerkiksi suhteellisuusteorian mukainen korjaus satelliittien kelloihin, sillä virallinen GPS-aika maanpinnalla ja satelliittien aika eroavat toisistaan satelliittien ja maanpinnan nopeuserojen johdosta. (Seeber 2003, s. 297-300.)

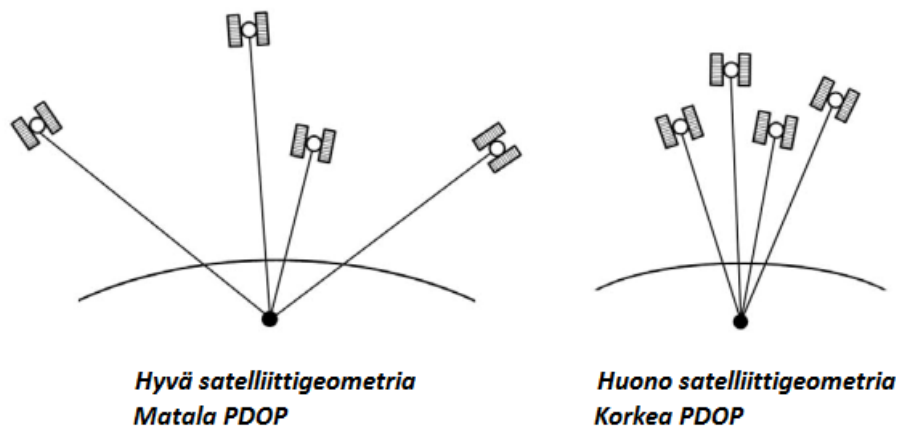
Signaaliin ilmakehässä vaikuttavia virheitä korjataan mallintamalla ionosfäärin (80-400 km) ja troposfäärin (0-15 km) ominaisuudet mahdollisimman tarkasti käyttäjän sijainnissa. Laskennassa käytettävissä olevat mallit ratkaisevat siis ilmakehän signaaliin aiheuttaman hidastumisen arvion eli käytännössä virheen suuruuden. Vastaanottimet, jotka pystyvät havainnoimaan useampia eri signaalitaajuuksia, esimerkiksi GPS:n tapauksessa L1-taajuutta ja uudempien satelliittien L2- ja L5-taajuuksia, pystyvät käytännössä eliminoimaan ionosfäärin aiheuttamat virheet paikannukseen, sillä eri aallonpituuden signaalit kulkevat eri tavoin ionosfäärissä (Mehtälä 2012). Troposfäärin, jossa paikalliset sääilmiöt tapahtuvat, virheitä on vaikeampi mallintaa ja korjata.

Vastaanottimella vaikuttavat virheet ovat sijainnin laskentaan ja havainnointikohinaan liittyviä virheitä, joiden suuruus riippuu laitteiston suorituskyvystä ja käytettyjen algoritmien tehokkuudesta. Toinen merkittävä virhelähde vastaanottimella on monitieheijastumaksi kutsuttu ilmiö, joka tarkoittaa signaalin reflektoitumista vastaanottimeen jonkin fyysisen kappaleen kautta. Esimerkiksi paljon lasia sisältävien kerrostalojen julkisivut voivat aiheuttaa runsaasti monitieheijastumia läheisyyteensä. Kuvassa 2 on havainnollistettu erilaisten monitieheijastumien syntymistä.



Kuva 2: Monitieheijastumat satelliittipaikannuksessa (Seeber 2003).

Paikannustarkkuuteen vaikuttaa UERE:n lisäksi satelliittien geometria taivaankannella suhteessa vastaanottimeen, mikä aiheuttaa virheeseen multiplikatiivisen vaikutuksen. Satelliittigeometriasta johtuvan epätarkkuuden kuvailuun käytetään termiä DOP (Dilution of Precision). Erilaisia DOP-arvoja ovat TDOP aikaepävarmuudelle, PDOP 3D-sijaintitarkkuudelle, HDOP ja VDOP horisontaaliselle ja vertikaaliselle epävarmuudelle vastaavasti sekä GDOP jossa aika- ja sijaintitarkkuus on yhdistetty. Satelliittigeometria on hyvä silloin, kun satelliitit ovat levittyneet tasaisesti taivaalle ja huono mikäli satelliitit ovat samassa tasossa vastaanottimeen nähden. Kuvassa 3 on havainnollistettu satelliittigeometrian PDOP:n yhteyttä. Yleisimmin satelliittipaikannuksen luotettavuuden arviointiin mittaushetkellä käytetään PDOP-arvoa. PDOP voidaan määrittellä sijaintiratkaisun laskennan yhteydestä saatavasta varianssi-kovarianssimatriisista (Seeber 2003).



Kuva 3: Satelliittigeometria ja PDOP (Seeber 2003).

PDOP-arvoilla 0-5 saadaan hyviä tuloksia, 5-10 kohtalaisia, ja tätä suuremmilla on odotettavissa suuria epätarkkuuksia (Poutanen 2016, s. 231).

Seeberin (2003) mukaan satelliittien määrän kasvaminen GNSS-järjestelmissä on kuitenkin vähentänyt DOP:n merkitystä paikannustarkkuuteen, sillä vastaanotin havaitsee lähes aina useiden satelliittien signaaleja ja satelliittigeometria säilyy lähes aina hyvänä. Kuitenkin esimerkiksi navigointisovelluksissa, joissa näkyvyys osaan satelliiteista voi hetkellisesti estyä rakennusten tai kasvillisuuden takia, DOP on edelleen hyödyllinen indikaattori paikannuksen luotettavuudelle.

### 3.1.3 Differentiaalinen paikannus

Differentiaalinen paikannus (DGPS, nykyään joskus kutsuttu myös DGNSS:ksi) ei oleellisesti eroa tekniikaltaan absoluuttisesta paikannuksesta. Erona on se, että tunnetulla tukiasemalla sijaitseva vastaanotin pyrkii havainnoimaan samoja satelliitteja kuin käyttäjä. Näiden havaintojen perusteella ionosfäärin ja troposfäärin aiheuttamat viipeet signaalin kulkuun sekä satelliitin rata- ja kellovirheet on mahdollista ratkaista tarkemmin kuin absoluuttisessa paikannuksessa. Tukiasema välittää korjausdataa käyttäjän vastaanottimeen, eli tekniikka vaatii tietoliikenneyhteyden tukiasemalle ja ohjelmiston, joka pystyy hyödyntämään korjausdataa. Differentiaalisella paikannuksella vastaanottimen sijainti on mahdollista määrittää noin 0,5–3 metrin tarkkuudella. (Seeber 2003, s. 326-335.)

Maanmittauslaitoksen Paikkatietokeskus tarjoaa DGPS-korjausdataa ilmaiseksi internet-yhteyden välityksellä NTRIP-protokollan (Networked Transport of RTCM via Internet Protocol) mukaisesti. Korjausdata perustuu FinnRef-tukiasemaverkoston tukiasemien laskemaan korjausdataan ja toimii sitä paremmin, mitä lähempänä tukiasemia paikannus tapahtuu. Mitausepävarmuus kasvaa noin 20–60 cm / 100 km etäisyyden tukiasemaan kasvaessa. Pääkaupunkiseudulta katsoen lähin tukiasema on Metsähovin tukiasema Kirkkonummella. FinnRef-tukiasemaverkkoa käyttäen paikantamisen pitäisi onnistua likimain 0,5 m tarkkuudella kaikkialla Suomessa. (Poutanen 2016, s. 257-259, MML Paikkatietokeskus.)

### 3.1.4 Tarkemmat menetelmät

Satelliittipaikannuksen avulla on mahdollista tehdä mittauksia myös senttimetriluokan tarkkuudella. Useimmissa senttimetriluokan tarkkuutta vaativissa mittauksissa käytetään RTK-mittauksia (Real Time Kinematic), jossa sijoitetaan yksi vastaanotin tunnetulle pisteelle tukiasemaksi ja suoritetaan varsinainen mittaus liikkuvalla vastaanottimella. Kumpikin vastaanotin seuraa samoja satelliitteja ja paikannusratkaisu perustuu vastaanottimien välisen vektorin ratkaisuun. Tavanomaisen C/A-koodiin perustuvan ratkaisun sijaan etäisyydet satelliitteihin ratkaistaan lisäksi kanta-aallon vaihehavaintojen perusteella. Tämän johdosta etäisyys satelliitteihin saadaan laskettua tarkasti. RTK-mittauksella saavutetaan senttimetriluokan tarkkuuksia. Nykyisin kaupalliset toimijat kuten Trimble ja Leica tarjoavat Suomessa RTK-tukiasemaverkkoihin perustuvia satelliittipaikannuspalveluita (Geotrim, Leica Geosystems). RTK-paikannusta tukiasemaverkon avulla kutsutaan NRTK:ksi (Network RTK). Erilaiset NRTK-palvelut toimivat hieman eri periaattein – tukiasemaverkosta voidaan esimerkiksi interpoloida virtuaalinen RTK-tukiasema käyttäjän lähelle, jolloin käyttäjälle riittää yhden vastaanottimen käyttö. RTK-mittauksia virtuaalista tukiasemaa käyttäen kutsutaan VRS-mittaukseksi (Virtual Reference Station). (Seeber 2003, s. 336-344; Poutanen 2016 s. 265.)

Satelliittipaikannuksella on mahdollista päästä vielä tätäkin tarkempaan mittaukseen, mutta nämä menetelmät nojaavat staattiseen mittaukseen ja vastaanotinvierokkojen havaintojen jälkilaskentaan ja rajautuvat siten pois liikennetelematiikan sovelluksista. Sama pätee myös PPP-tekniikkaan (Precise Point Positioning), jota käyttämällä on mahdollista saavuttaa senttimetriluokan paikannustarkkuus yhdellä vastaanottimella. PPP vaatii korjatut satelliittien rata- ja kellovirhetiedot, joita tarjoaa esimerkiksi IGS (International GNSS Service). IGS:n korjausdataa voi käyttää myös reaaliaikaisesti, mutta PPP-mittauksen pitkät konvergenssiajat (alustukseen kuluva aika saattaa olla jopa tunteja) ja häiriöalttius estävät sen käytön ajoneuvopaikantamiseen (Mehtälä 2012, Poutanen 2016 s. 268-269).

### 3.2 Merkintälasku

Merkintälaskulla (kutsuttu myös vektoripaikannukseksi, engl. dead reckoning, DR) tarkoitetaan suhteellista paikannusta, jonka avulla ajoneuvon sijainti on mahdollista määrittää suhteessa tunnettuun lähtöpisteeseen. Sensoreita, joita merkintälaskuun perustuvat järjestelmät käyttävät, ovat esimerkiksi gyroskoopit, joiden avulla voidaan mitata ajoneuvon rungon suunnan muutoksia sekä ajoneuvon odometri, joka määrittää nopeuden mittaamalla ajoneuvon renkaan pyörähdyskertoja. Nämä tiedot yhdistämällä tiedetään, kuinka pitkän matkan ja mihin suuntaan ajoneuvo on kulkenut. Myös inertiamittaukseen perustuva merkintälasku on mahdollista, jolloin mitataan ajoneuvon kiihtyvyyksiä eri suuntiin. (Quddus 2006.)

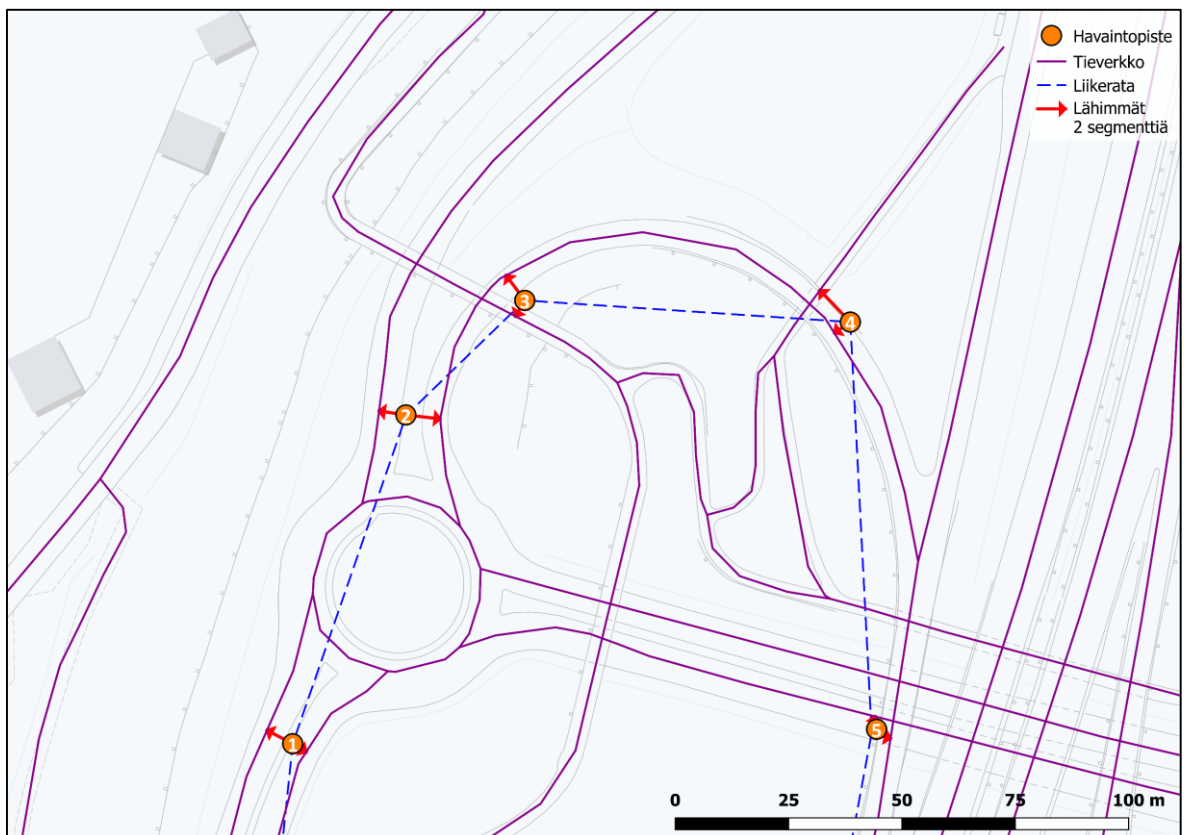
Edulliset pietsosähköiset MEMS-gyroskoopit (Micro-electro-mechanical system) ovat lisänneet merkintälaskuun perustuvan paikannuksen houkuttelevuutta 2000-luvulla, ja niitä onkin asennettu esimerkiksi moniin nykyisin markkinoilla oleviin älypuhelimiin. Tällaisissa gyroskoopeissa on kuitenkin huomattavia epävarmuuksia. Merkintälaskun yleisenä heikkoutena on sen eri virheiden kumuloituminen lopulliseen sijainnin määrittämiseen, mistä seuraa ajan myötä harhautuminen yhä kauemmas todellisesta sijainnista (Zhao et al. 2003).

Satelliittipaikannus ja merkintälaskumenetelmät on mahdollista yhdistää integroiduksi paikannusratkaisuksi, joka kompensoi molempien menetelmien puutteita. Satelliittipaikannuksella saadaan jatkuvasti absoluuttisia sijaintiratkaisuja integroituun järjestelmään, mikä korjaa merkintälaskun virheiden kumuloitumisongelman ja harhautumisen. Merkintälasku puolestaan tarjoaa tarkkoja sijaintiratkaisuja silloin, kun satelliittipaikannus on mahdotonta tai sen DOP on korkea heikon satelliittigeometrian tai näkyvyyden johdosta (Zhao et al. 2003). Satelliittipaikannuksen ja merkintälaskun yhdistäviä integroituja ratkaisuja kutsutaan tässä työssä integroiduiksi GPS/DR- tai GNSS/DR-paikannusjärjestelmiksi.

Zhao et al. (2003) toteuttivat integroidun paikannusjärjestelmän, joka hyödynsi tavallista yksitaajuuksista GPS-vastaanotinta ja edullista MEMS-gyroskooppia. Järjestelmässä käytettiin Kalman-suodinta yhdistämään satelliittipaikannuksen ja DR-sensoreiden data. Kalman-suodin on algoritmi, jota käytetään dynaamisten järjestelmien tilojen estimointiin ja ennustamiseen rekursiivisesti kohinaisen sensoridatan perusteella. Järjestelmän etuna oli satelliittipaikannuksen ongelmien korjaaminen erityisesti urbaanissa ympäristössä, kuten korkean ja tiiviin rakentamisen alueilla (niin sanotuissa urbaaneissa kanjoneissa) sekä tunneleissa. Tällaisissa ympäristöissä satelliittipaikannus saattaa olla mahdotonta tai altista virheille.

## 4 Karttasovitus

Karttasovitus tarkoittaa ajoneuvon liikeradan sovittamista digitaaliseen tieverkkoaineistoon. Liikerata koostuu pistemäisistä havainnoista, joista jokaisella on aikaleima ja sijainnin määrittävä koordinaattipari. Yleensä liikeradat saadaan kerättyä GNSS-paikannuksella tai integroidulla GNSS/DR-paikannuksella. Sovelluksesta ja käytettävistä menetelmistä riippuen karttasovituksen tuloksena voidaan saada pisteitä, joiden sijaintia on korjattu siten, että ne sijoittuvat tieverkolle ajoneuvon mittaushetken todennäköiseen todelliseen sijaintiin, tai toisaalta kokonaisia reittejä, jotka kuvaavat ajoneuvon todennäköisintä reittiä tieverkolla. Karttasovitusalgoritmin tärkein ominaisuus on useimmissa sovelluksissa sen tieverkon segmentin tunnistaminen, jolla ajoneuvo todellisuudessa on havainnon hetkellä (Zhao et al. 1997). Toissijaisena tavoitteena pidetään tietyissä sovelluksissa esimerkiksi ajoneuvon sijainnin tarkempaa määrittystä segmentillä, kuin mikä pelkistä havaintopisteistä olisi pääteltävissä (Quddus 2006). Koska data on kohinaista ja sisältää virheitä luvussa 3.1.2 kuvattujen seikkojen johdosta ja lisäksi data on usein telematiikkasovelluksissa melko harvaa (liikeradan pisteiden aikaväli on 15 s – 1 min tai jopa useita minuutteja), yksinkertaiset tieverkon lähimmän segmentin haut tuottavat paljon virheitä. Urbaanit ympäristöt tuovat lisää haasteita karttasovitukseen niiden GNSS-paikannukselle aiheuttamien erityishaasteiden johdosta. Lisäksi tieverkko on tällaisissa ympäristöissä tiheämpi, jolloin eri katujen segmentit ovat hyvin lähelläkin ja voivat kulkea rinnakkain samansuuntaisesti. Kuvassa 4 on havainnollistettu karttasovituksen ongelmaa. Lähimmän segmentin haku tuottaa tällaisessa hyvin yleisessä tapauksessa virheellisiä tuloksia. Liikeradan pisteet 2, 3 ja 5 sovitettaisiin kuvassa väärille segmenteille.



Kuva 4: Karttasovitusongelma.



Karttasovitusalgoritmeja voidaan luokitella muutamilla eri tavoilla. Niitä voidaan eritellä ensinnäkin sen perusteella, onko algoritmi inkrementaalinen vai globaali (englanninkielisessä kirjallisuudessa esiintyy usein global/incremental tai online/offline). Globaalit algoritmit vaativat koko liikeradan tiedon, sillä myös liikeradassa myöhemmin havaitut pisteet vaikuttavat niitä edeltävien pisteiden sovitukseen, jolloin lopullinen sovitus voidaan tehdä vasta jälkilaskentana, kun ajoneuvon liikerata on havainnoitu kokonaan. Inkrementaaliset algoritmit toimivat vain sillä datalla, mitä on kullakin hetkellä saatavissa, eli yksittäinen havaintopiste ja mahdollisesti sitä edeltävät pisteet. Esimerkiksi ajonaikaiset navigointisovellukset vaativat usein käytännössä inkrementaalisten algoritmien käyttöä. Ajonaikaisissa sovelluksissa algoritmien suorituskyvylle on enemmän vaatimuksia kuin useissa muissa sovelluksissa, sillä niiden on toimittava lähes reaaliaikaisesti ja usein laskennallisesti rajoittuneissa ympäristöissä – yleensä kännykkäsovelluksissa tai autonavigaattoreissa (Jagadeesh et al. 2004). Globaaleja algoritmeja puolestaan suoritetaan usein esimerkiksi keskitetyllä palvelimella jälkilaskentana. Toisaalta monista globaaleista algoritmeista on kehitetty muunnelmia, joiden johdosta niitä voidaan suorittaa inkrementaalisesti ja päivittää koko laskettua tulosta uutta dataa saataessa. (Lou et al. 2009, Goh et al. 2012.)

Quddus (2006) luokittelee väitöskirjassaan karttasovitusalgoritmeja niiden toteutustavan mukaan geometrisiin, topologisiin ja kehittyneisiin algoritmeihin.

*Geometriset algoritmit* ovat yksinkertaisia ja laskennallisesti tehokkaita ja ne hyödyntävät pääasiassa havaitun pisteen etäisyyttä tieverkkoon. Geometriset algoritmit voidaan jakaa piste–piste-, piste–kaari- ja kaari–kaari-algoritmeihin. Piste–piste-algoritmeissa valitaan tieverkosta lähin piste ja sovitetaan havainto siihen. Piste–kaari-algoritmeissa puolestaan etsitään verkon lähin segmentti, ja projisoidaan havainto sille. Kaari–kaari-algoritmeissa verrataan koko havaittua liikerataa tieverkon kaariin. Esimerkiksi White et al. (2000) sovelsivat kaari–kaari-vertailua hakemalla tieverkon pisteitä havaintopisteiden läheisyydestä, ja muodostivat näistä pistekandidaateista kaarikandidaatteja, joihin havaitun liikeradasta muodostettua kaartia verrattiin, ja lähintä kaarikandidaattia vastaava tieverkon kaari valittiin todennäköiseksi reitiksi. (Quddus 2006.)

*Topologiset algoritmit* pyrkivät pelkän etäisyyden lisäksi hyödyntämään tieverkon topologiaa sovituksessa. Geoinformatiikassa topologialla tarkoitetaan spatiaalisten kohteiden välistä topologisia riippuvuuksia, kuten vierekkäisyyttä, yhdistyvyyttä ja leikkaavuutta. Topologisissa algoritmeissa tutkitaan tieverkon yhdistyvyyttä ja pyritään löytämään oikeat pistekandidaatit tämän perusteella. Topologiset algoritmit ovat pääsääntöisesti parempia kuin geometriset algoritmit ja lisäksi laskennallisesti tehokkaita ja sopivat siten myös reaaliaikaisiin sovelluksiin. (Quddus 2006, Greenfeld 2002.)

*Kehittyneiksi algoritmeiksi* Quddus (2006) laskee sellaiset karttasovitusalgoritmit, jotka hyödyntävät jotakin kehittyntä matemaattista konseptia tai menetelmää. Tällaisia algoritmeja on Qudduksen mukaan esimerkiksi Kalman-suodinta, Dempster-Shaferin uskottavuusteoriaa, sumeaa logiikkaa tai Bayesilaista päättelyä hyödyntävät algoritmit.

## 4.1 Kirjallisuuskatsaus algoritmeista

Karttasovitusalgoritmeja on kehitetty ainakin 1970-luvulta alkaen (French ja Lang 1973). 1970- ja 1980-luvuilla niitä käytettiin DR-sensoreiden harhautumisen korjaamiseen eli ajoneuvonavigointiin ennen GPS-aikaa, tai ennen kuin satelliittipaikannuksen kattavuus oli tarpeeksi suuri tieliikenteen navigoinnin tarpeisiin. Karttasovitusmenetelmät olivat yleensä yksinkertaisia geometrisia tieverkon lähimmän pisteen tai lähimmän segmentin hakuja sellaisissa kohdissa, joissa DR-sensoreiden perusteella havaittiin ajoneuvon kääntyminen (French 1986, Collier 1990). Edelleen 1990-luvulla geometrisia menetelmiä käytettiin GPS-paikannuksen karttasovituksessa (Bernstein ja Kornhauser 1996, White 2000). Kehittyneitä menetelmiä hyödyntävät algoritmit keskittyivät 1980- ja 1990-luvuilla etenkin Kalman-suotimeen perustuvan karttasovituksen kehittämiseen DR-sensoreiden tai myöhemmin integroidun GPS/DR-paikannuksen paikannustarkkuuden parantamiseksi (Krakiwsky et al. 1988, Tanaka et al. 1990).

Varhaisista kehittyneistä menetelmistä kannattaa mainita ainakin Scott (1994), joka käytti integroitua GPS/DR-paikannusta Kalman-suodinta hyödyntäen sekä karttasovitusta Bayesilaisella MAP-estimaattorilla (maksimi *a posteriori*) – kuitenkin edelleen ainoastaan yksinkertaisia geometrisia suureita ja Kalman-suotimelta saatavia virhesuureita hyödyntäen. Kalman-suotimen käyttöä ajonaikaisissa navigointisovelluksissa karttasovituksen tukena on käytetty ja tutkittu myös tästä eteenpäin erityisesti silloin, kun on hyödynnetty integroitua GNSS/DR-paikannusta, mutta myös kohinan poistamiseen satelliittipaikannusdatasta (Kim et al. 2000, Zhao et al. 2003, Najjar ja Bonnifait 2003, Ochieng et al. 2004, Szwed ja Pekala 2013).

2000-luvulla kirjallisuudessa keskityttiin paljolti itsenäisen GNSS-paikannuslaitteen havaintodatan karttasovitukseen. Tähän suuntaukseen vaikutti todennäköisesti erityisesti se, että GPS-järjestelmän julkiseen paikannuskoodiin tehty tahallinen SA-heikennys poistettiin vuonna 2001. 2000-luvun alkupuolella julkaistiin lukuisia tutkimuksia, joissa hyödynnettiin topologisia karttasovitusalgoritmeja.

Greenfeld (2002) esitteli artikkelissaan tällaisen topologisen algoritmin. Kaksiosaisen topologisen algoritmin ensimmäisessä osassa suoritettiin algoritmin alustus hakemalla ensimmäinen karttasovitettu segmentti – käytännössä lähin tieverkon piste. Toisessa osiossa karttasovitusta jatkettiin siten, että mahdollisten segmenttikandidaattien joukko rajautui niihin tieverkon kaariin, jotka yhdistyvät ensimmäiseen. Suuri osa topologisista algoritmeista nojaa tämän kaltaiseen moniosaiseen algoritmiin (esimerkiksi Yu et al. 2002, Xu et al. 2002, Li ja Fu 2003, Velaga et al. 2009). Tällaisten algoritmien yhteisenä heikkoutena voidaan pitää karttasovituksen onnistumisen riippuvuutta ensimmäisen pisteen sovituksesta alustusvaiheessa. Jos ensimmäinen piste sovitetaan väärin, ovat loputkin liikeradan pisteet suurella todennäköisyydellä vääriä (Quddus et al. 2006).

Topologisissa algoritmeissa tutkitaan yleensä vain edelliseen karttasovituksessa valittuun segmenttiin yhdistyviä seuraavia segmenttejä. Jos yhdistyvien segmenttien joukosta ei löydy hyviä kandidaatteja, palautuu algoritmi alustusvaiheeseen. Tämän ominaisuuden johdosta algoritmit sopivat vain melko tiheään paikannusfrekvenssin sovelluksiin. Jos liikeratadata on harvaa, osalla tieverkon segmenteistä ajoneuvon todellisella reitillä ei ole välttämättä yhtäkään havaintoa (Quddus ja Ochieng 2014.)

Najjar ja Bonnifait (2003) hyödynsivät Dempster-Shaferin uskottavuusteoriaa (DST) ja Kalman-suodinta karttasovituksessa. DST:n perusideana on antaa erilaisille vaihtoehdoille positiivisia todennäköisyyksiä välillä  $[0.0, 1.0]$ , joita kutsutaan DST:ssa uskottavuuksiksi. DST:ssa voidaan joustavasti yhdistää epävarmaa, heterogeenistä tietoa useista eri lähteistä, esimerkiksi asiantuntijatietoa, tai karttasovituksen tapauksessa tietoa erilaisista sensoreista tai tieverkkoon liittyvistä suureista. Kirjoittajat käyttivät algoritminsa muuttujina havaintopisteen suuntimaa ja etäisyyttä tieverkon segmentteihin nähden. Muita DST:ta hyödyntäviä karttasovitusmenetelmiä ovat käyttäneet Yang et al. (2003) ja Srinivasan ja Chey (2003).

1990- ja etenkin 2000-luvulla kehitettiin useita algoritmeja, jotka pyrkivät ratkaisemaan karttasovitusongelman sumeaa logiikkaa käyttäen. Sumeassa logiikassa pyritään Boolean logiikan (*tosi, epätosi*) sijaan mallintamaan osittaista totuutta antamalla muuttujille jäsenyysasteet järjestelmän sumeissa joukoissa (esimerkiksi  $17\text{ }^{\circ}\text{C}$  ulkolämpötila voisi kuulua luokkaan viileä jäsenyysasteella 0,3 ja luokkaan lämmin jäsenyysasteella 0,7). Tavanomaisessa järjestelmässä diskreetit muuttujat sumeutetaan sumeiden jäsenyysfunktioiden perusteella, jolloin niille voidaan suorittaa regressiota tai luokittelua sumein laskusäännöin. Lopputuloksena saatava sumea tulos yleensä selkeytetään jollakin selkeytysfunktioilla, jolloin lopputulokseksikin saadaan diskreetti lukuarvo (Zadeh 1973).

Kim ja Kim (2001) toteuttivat sumeaa logiikkaan nojautuvan karttasovitusalgoritmin, jonka muuttujina olivat havainnon etäisyys tieverkosta, suuntima tieverkon segmentteihin, sekä segmentin edellisen pisteen sovituksessa saama sumea pisteytys  $C$ . Algoritmi oli kaksiosainen sisältäen karttasovituksen alustusvaiheen, seurantavaiheen ja lisäksi risteysalueen tunnistuksen. Alustusvaiheessa tutkittiin kaikkia segmenttikandidaatteja ja valittiin niistä korkeimman sumean pisteytyksen saanut segmentti seurantaan, mikäli  $C$  ylitti raja-arvon  $C_t$ . Seurantavaiheessa ei tutkittu mitään muuta kuin seurattavaa segmenttiä ja päivitettiin sen  $C$ -arvoa suhteessa kahteen raja-arvoon: alustuksen raja-arvon  $C_t$  alittaminen aiheutti paluun alustustilaan ja risteysraja-arvon  $C_j$  alittaminen yhdistettynä pieneen etäisyyteen seuraavasta risteyksestä aiheutti siirtymisen risteysalueen tunnistukseen. Risteysalueella haettiin kaikki segmentit, jotka yhdistyivät edelliseen seurattuun segmenttiin ja alustettiin niiden pisteytykseksi edellisen segmentin viimeisin pisteytys. Sen jälkeen kaikkien kandidaattien pisteytykset päivitettiin perustuen taas etäisyyteen ja suuntimaan, ja niistä korkeimman pisteytyksen saanut segmentti valittiin seurattavaksi segmentiksi.

Quddus et al. (2006) käyttivät Kalman-suotimen ja GPS/DR-paikannuksen kanssa kolmi-osaista sumeaa algoritmia, jossa eri vaiheet olivat alustus, seuranta ja segmentin valinta risteyksessä. Alustuksessa hyödynnettiin ajoneuvon nopeutta, suuntimaa ja etäisyyttä segmenttikandidaatteihin sekä GPS-paikantimen havainnolle antamaa HDOP-arvoa. Segmentin seurannassa käytettiin lisäksi segmentillä kuljetun matkan pituutta suhteessa segmentin pituuteen, sekä ajoneuvon kulkusuunnan muutosnopeutta ja sijaintihavainnon kulmaeroja suhteessa segmentin alku- ja loppupisteisiin. Kuljetun matkan pituuden ja kulmaerojen perusteella saatiin selville, milloin ajoneuvo oli ohittanut segmentin lopun eli päätynyt risteykseen. Uuden seurattavan segmentin valitsemiseksi risteyksissä tutkittiin etäisyyttä segmenttikandidaatteihin ja niiden topologista yhdistyvyyttä edelliseen segmenttiin. Artikkelissa raportoitiin erittäin korkea 99,2 % onnistumisprosentti oikeiden segmenttien tunnistukselle. Artikkelissa implementoitiin myös yksinkertaiset geometriset menetelmät, joita Bernstein ja Kornhauser (1996) ja White (2000) ehdottivat, topologinen menetelmä, jota Greenfeld

(2002) ehdotti ja sen kehitelty versio (Quddus 2003) sekä Kalman-suodinta ja GPS/DR-paikannusta hyödyntävä topologinen menetelmä (Ochieng 2004). Vertailun tulokset on listattu taulukossa 2.

| <i>Algoritmi</i>                              | <i>Tiesegmentin oikea tunnistus</i> |
|---|-------------------------------------|
| Piste–piste<br>(Bernstein ja Kornhauser 1996) | 70,5 %                              |
| Piste–kaari<br>(White 2000)                   | 76,8 %                              |
| Topologinen<br>(Greenfeld 2002)               | 90,3 %                              |
| Topologinen<br>(Quddus 2003)                  | 95,4 %                              |
| Topologinen / Kalman-suodin<br>(Ochieng 2004) | 98,1 %                              |
| Sumea logiikka<br>(Quddus 2006)               | 99,2 %                              |

*Taulukko 2: Karttasovitusalgoritmien vertailu (Quddus et al. 2006).*

Myös joukko muita sumeaa logiikkaa karttasovitukseen hyödyntäviä artikkeleita on julkaistu, esimerkiksi Jagadeesh et al. (2004), Syed ja Cannon (2004), Fu et al. (2004). Kaikki käyttävät joitain sekoituksia edellä mainituista menetelmistä ja muuttujista, sekä usein riisteysalueen tunnistamista ja yksinkertaista yhdistyvyyden tutkimista segmentiltä toiselle siirryttäessä.

Kehittyneemmistä geometrisista algoritmeista voidaan vielä mainita Fréchet-etäisyyteen perustuvat kaari–kaari-vertailut. Fréchet-etäisyys tarkoittaa hetkellistä käyrien välistä minimietäisyyttä, jolla käyriä pitkin kuljettaessa voidaan pysytellä, siten että kulkusuunta pysyy samana. Alt et al. (2003) esittelivät ensimmäisenä menetelmän soveltamista karttasovitukseen. Brakatsoulas et al. (2005) kehittivät menetelmää edelleen ja totesivat että niin sanottu heikko Fréchet-etäisyys, joka on laskennallisesti yksinkertaisempi toteuttaa, sopii karttasovitukseen yhtä hyvin kuin varsinainen Fréchet-etäisyyskin. Chen et al. (2011) kehittivät Fréchet-etäisyyden laskentaa karttasovitusongelmassa yleistämällä verrattavia käyriä ja he saavuttivat suuria suorituskykyparannuksia verrattuna aikaisempiin menetelmiin. Lou et al. (2009) sovittivat oikealle tiesegmentille lähes 90 % pisteistä testatessaan Fréchet-etäisyyteen perustuvaa algoritmia harvalla liikeratadatatalla. Wei et al. (2012) testasivat eri karttasovitusalgoritmeja ja saavuttivat Fréchet-etäisyyteen perustuvalla karttasovituksella noin 97 % tarkkuuden 8 s havaintovälillä ja noin 92 % tarkkuuden 16 s havaintovälillä.

Lou et al. (2009) kehittivät uudenlaisen globaalin algoritmin, joka otti huomioon havaintojen etäisyyden segmenttikandidaatteihin nähden ja nopeuden yhteensopivuuden tieverkon nopeusrajoituksiin. Lisäksi topologia otettiin huomioon laskemalla etukäteen pistekandidaattien väliset lyhyimmät reitit ja vertaamalla niiden pituuksia  $l$  pisteiden välisiin euklidisiin etäisyyksiin  $l_0$  kaavalla  $p = l_0/l$ . Pistekandidaatit mallinnettiin tämän jälkeen verkkona, jonka kaarien joukko sisälsi kaikki mahdolliset siirtymät pistekandidaattien välillä. Kaaren painoksi tuli yhdistelmä kaikista edellä mainituista muuttujista. Lopuksi optimaalinen reitti etsittiin verkosta maksimoimalla reitin pituus ja kuljetun reitin painot. Kirjoittajat kutsuivat algoritmia nimellä ST-matching (Spatio-Temporal matching).

Yuan et al. (2010) pyrkivät kehittämään Lou et al. (2009) ST-matching algoritmia. Heidän perusajatuksenaan algoritmin kehityksessä oli, että kaikki pisteet vaikuttavat toisiinsa, mutta läheiset pisteet enemmän kuin kauemmat. Algoritmi laski todennäköisyydet pisteille ja niiden siirtymille samalla tavalla kuin ST-matching-algoritmikin, mutta todennäköisyydet määriteltiin matriisiin, jota kutsuttiin artikkelissa staattiseksi pisteytysmatriisiksi. Lisäksi jokaiselle havaintopisteelle laskettiin etäisyysmatriisi, jonka arvoja käytettiin painoina määriteltäessä pisteiden vaikutusta toisiinsa. Havaintopisteille muodostettiin omat painotetut pisteytysmatriisit, jotka muodostuivat staattisen pisteytysmatriisin ja havaintopisteiden etäisyysmatriisien tuloina. Algoritmi kävi läpi jokaisen pistekandidaatin ja haki sen painotetusta pisteytysmatriisista sellaisen optimaalisen reitin, joka kulkee kyseisen pistekandidaatin kautta. Samalla pistekandidaatti antoi äänen kaikille pisteille, jotka olivat sen lokaalin optimaalisen reitin osia. Lopuksi muodostettiin reitti, joka kulkee kaikkien eniten ääniä per havaintopiste saaneiden pistekandidaattien kautta. Algoritmia kutsuttiin nimellä IVMM (Interactive Voting-Based Map Matching) ja se saavutti kirjoittajien mukaan 30 s havaintovälillä noin 85 % sovitustarkkuuden.

Quddus ja Ochieng (2014) toteuttivat karttasovitusalgoritmin harvalle liikeratadatalle hyödyntäen lyhyimpien reittien laskentaa, kuten edellä mainitut artikkelit käyttäen muuttujina lisäksi pistekandidaattien suuntima- ja etäisyystietoja, sekä suunnanmuutosta edelliseen pisteeseen verrattuna. Parametrien painot määriteltiin optimoimalla ne geneettisellä algoritmilla verrattuna tarkalla RTK-paikannuksella mitattuun referenssiaineistoon. Geneettisellä algoritmilla tarkoitetaan optimointimenetelmää, joka jäljittelee evoluutiota luomalla populaatioita, risteyttämällä niitä ja sallimalla mutaatioiden syntymisen ja pisteyttämällä uusien populaatioiden elinkelpoisuus. Geneettisillä algoritmeilla voidaan optimoida järjestelmiä, jotka sisältävät useita heterogeenisiä muuttujia, mikäli onnistutaan kehittämään sopiva pisteytysfunktio (Thede 2004). Kirjoittajat optimoivat lisäksi reitinhakuun käytettyä aikaa käyttämällä A\*-algoritmia, joka lopetti reittien laskennan aina, mikäli niiden pituus kasvoi yli 1000 metriin lähtöpisteeseen verrattuna. Tästä seurasi huomattavia nopeusparannuksia sellaisissa tapauksissa, joissa reittiä ei voida löytää. Artikkelissa raportoitiin korkea, 98-99 % sovitustarkkuus, joka pysyi samalla tasolla kaikilla havaintoväleillä yhdestä sekunnista 30 sekuntiin asti.

2010-luvulla karttasovituksesta kirjoitetut artikkelit ovat keskittyneet suurelta osin Markovin piilomalliin (HMM, Hidden Markov Model) perustuviin globaaleihin algoritmeihin. HMM on aikasarjadatan mallintamiseen sopiva menetelmä, jossa pyritään estimoimaan mallin tilat aikasarjan kuluessa. Tilat ovat kuitenkin tuntemattomia ja niistä saadaan tietoon vain jokin välillinen havainto, jota kutsutaan emissioksi. Havaintotodennäköisyydet määritellään emissiomatriisiin, joka on havainnon diskreetti todennäköisyysjakauma kaikille mallin eri tiloille. Lisäksi mallin tiloille määritellään transitiotodennäköisyydet transitiomatriisiin, joka sisältää todennäköisyyden siirtymälle kaikkien mallin eri tilojen välillä. Mikäli HMM:sta halutaan löytää yksittäinen todennäköisin tilasekvenssi, voidaan käyttää dynaamiseen ohjelmointiin perustuvaa Viterbi-algoritmia, jolla voidaan etsiä todennäköisin tilojen sekvenssi, kun on tehty tietty havaintosekvenssi. HMM:iin perustuvat karttasovitusalgoritmit pyrkivät siis mallintamaan havaintopisteen sopivuuden tieverkon segmenttikandidaateille emissioina, ja siirtymät pisteiden välillä tieverkossa transitioina. Viterbi-algoritmeilla haetaan tämän jälkeen liikeradalle todennäköisin karttasovitettu reitti tieverkossa.

Hummel (2006) käytti Markovin piilomallia karttasovituksen toteuttamiseen. HMM:n emissiotodennäköisyydet määriteltiin ajoneuvon etäisyydestä ja suuntimasta tieverkon segmenttikandidaateihin ja transitiotodennäköisyydet tieverkon topologiasta, siten että yhdistyvät segmentit saivat  $1/n$  transitiotodennäköisyyden, jossa  $n$  on yhdistyvien segmenttien lukumäärä. Tieverkon ajosuunnat otettiin huomioon mallinnuksessa siten, että väärään suuntaan yksisuuntaista segmenttiä kulkeva reitti saa hyvin pienen transitiotodennäköisyyden.

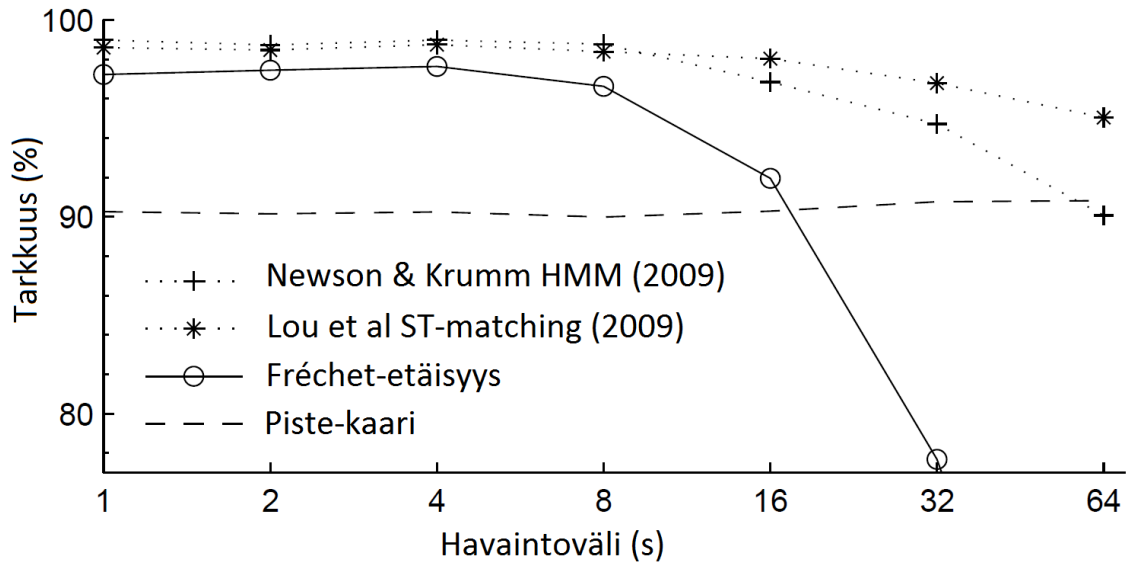
Newson ja Krumm (2009) käyttivät HMM:ia harvan liikeratadatan karttasovitukseen mallintamalla emissiotodennäköisyyksiksi havainnon etäisyydet segmenttikandidaateista olettamalla, että paikannuksen virhe on normaalijakautunut satelliittipaikannuksen virheen keskihajonnalla. Transitiotodennäköisyydeksi mallissa tulee kahden peräkkäisen epookin piste-kandidaattien välisen tieverkosta lasketun lyhyimmän reitin pituuden  $l$  ja pisteiden välisen isoympyräetäisyyden  $l_0$  välinen erotus. Isoympyräetäisyydellä tarkoitetaan kahden pisteen välistä lyhintä etäisyyttä maapallon pintaa pitkin. Käytännössä lyhyillä välimatkoilla voitaisiin hyvin käyttää myös euklidista etäisyyttä. Kirjoittajat havaitsivat, että etäisyysero  $l - l_0$  on oikein sovitetuilla reiteillä hyvin pieni, joten transitiotodennäköisyys voidaan mallintaa käyttäen eksponenttijakaumaa.

Ajoneuvon suuntima segmenttikandidaateihin jätettiin pois emissiotodennäköisyyden määrittämisestä sen johdosta, ettei artikkelin kirjoittajilla ollut käytössään muuta tietoa ajoneuvon kulkusuunnasta kuin kahden peräkkäisen havainnon muodostaman vektorin suuntakulma. Harvalla liikeradalla tämä ei ole karttasovituksen kannalta luotettavaa dataa. Reitin ja havaintopisteiden välisten pituuksien erot sopivat transitiotodennäköisyyden mallintamiseen harvalla liikeratadalla paremmin kuin suoraan verkosta saatava yhdistyvyystieto, sillä harvassa liikeradassa osalla segmenteistä ei ole lainkaan havaintopisteitä, ja tällaisissa tapauksissa seurattava liikerata pitäisi aina katkaista ja malli alustaa. Topologiatieto sisältyy tällaista menetelmää käytettäessä kuitenkin implisiittisesti transitiotodennäköisyyteen, sillä topologisesti epätodennäköiset siirtymät pisteiden välillä saavat suuren arvon tieverkosta lasketun reitin pituuden ja havaintopisteiden isoympyräetäisyyden erotukselle. Tästä seuraa alhainen transitiotodennäköisyys topologisesti epätodennäköiselle siirtymälle.

Virhepisteiden käsittelemiseksi pistekandidaattien etäisyyserot, jotka kasvoivat yli 2000 metriin aiheuttivat algoritmissa transitiotodennäköisyyden asettamisen nollassa. Toinen tarkistus, joka algoritmissa tehtiin, oli siirtymien maksiminopeuden asetus 180 kilometriin tunnissa. Transitiotodennäköisyys asetettiin nollassa, mikäli siirtymä pistekandidaattien välillä vaatisi tätä suuremman nopeuden. Jos kaikki kahden epookin väliset transitiotodennäköisyydet olivat 0, tehtiin algoritmissa erillinen mallin korjaus. Korjauksessa katkopisteen kummaltakin puolelta poistettiin yksi piste kerrallaan, kunnes malli korjaantui jälleen kulkukelpoiseksi. Mikäli mallin korjausyritykset kuitenkin kasvattivat kahden epookin välisen aikavälin yli 180 sekuntiin, tehtiin liikeradan katkaisu ja mallin alustus. Kirjoittajat raportoivat algoritmillemme erittäin korkeita yli 99 % sovitustarkkuuksia vertaamalla väärin sovitettujen reititiosuoksien pituuksia tunnetun reitin pituuteen.

Wei et al. (2012) toteuttivat HMM:ia hyödyntävän globaalin algoritmin, jonka muuttujat olivat samoja kuin Newsonin ja Krummin (2009) algoritmissa. He havaitsivat kuitenkin, että harvalla liikeratadalla (yli 16 s havaintoväli) Newsonin ja Krummin käyttämä transitiotodennäköisyyden määrittäminen absoluuttista reitin ja isoympyräetäisyyden eroa,  $l - l_0$ , käyttämällä aiheuttaa sovitustarkkuuden heikentymistä verrattuna Lou et al. (2009) käyttämään menetelmään, jossa tarkastellaan etäisyyksien suhteellista eroa  $l_0 / l$ . Saman havainnon teki myös

Koller et al. (2015) HMM-algoritmiaan käsitelleessä artikkelissa. Wei et al. optimoivat oman algoritminsä nopeutta lopettamalla reittien esilaskennassa käytettävän Dijkstran algoritmin reittivaihtoehtojen laskemisen, mikäli niiden nopeusvaatimus (eli reitin pituus suhteessa kahden havaintopisteen aikaleimojen eroon) kasvoi yli arvon 50 m/s. Kirjoittajat vertaavat artikkelissaan Lou et al. ST-matching algoritmia, Newsonin ja Krummin HMM-algoritmia, Fréchet-etäisyyden perustuvaa geometrista menetelmää ja yksinkertaista geometrista piste-kaari-menetelmää ja näiden suorituskyvyn muutosta erilaisilla havaintotiheyksillä. Tämän vertailun tulokset on esitetty kuvassa 5.



Kuva 5: Karttasovitusalgoritmien vertailu (Wei et al. 2012).

Goh et al. (2012) toteuttivat inkrementaalisen HMM-karttasovitusalgoritmin, jossa otettiin huomioon lukuisia muuttujia GPS/DR-paikannuslaitteella varustetussa ajoneuvossa. Tämän algoritmin emissiotodennäköisyydet muodostuivat etäisyydestä segmentteihin, mutta lisäksi huomioon otettiin tien leveys ja nopeusrajoitus. Ajatuksena oli se, että leveällä tiellä keskilinjan etäisyys ajokaistoista on suurempi, joten silloin sallitaan myös suurempi etäisyysero suhteessa havaintoihin, kuin pienellä tiellä. Transitiotodennäköisyyden määrittelyssä puolestaan verrattiin pistekandidaattien välisten lyhyimpien reittien pituutta DR-sensorilta saatuaan mitattuun matkaan. Tämän lisäksi tehtiin ylimääräinen pisteytysvähennys ajoneuvon kääntymisistä, eli algoritmi suosi suorempia reittejä. Algoritmissa hyödynnettiin adaptiivisen liukuvan ikkunan (Variable Sliding Window) menetelmiä ajonaikaisten tulosten laskemiseen inkrementaalisesti. Myös Szwed ja Pekala (2014) kehittivät inkrementaalista HMM-algoritmia, jossa mallia päivitetään dynaamisesti sitä mukaa, kun uusia havaintoja tehdään.

Song et al. (2012) kehittivät HMM-algoritmin, joka pystyy käyttämään moniytimisten prosessorien kapasiteettia hyödyntämällä rinnakkaisajoa tehokkaasti. He paransivat Newson ja Krumm (2009) algoritmia ottamalla huomioon nopeusrajoituksen emissiotodennäköisyyksissä. Lisäksi he havaitsivat, että Newsonin ja Krummin käyttämä eksponentiaalinen todennäköisyysjakauma transitiotodennäköisyyden laskemiseen, joka perustuu staattiseen  $\beta$ -parametrin arvoon, aiheuttaa ongelmia, mikäli liikeradan havaintoväli on vaihteleva. He esittivät artikkelissaan optimaaliset arvot  $\beta$ -parametrille erilaisilla havaintoväleillä.

Osogami ja Raymond (2013) pyrkivät parantamaan Newsonin ja Krummin (2009) algoritmia ottamalla huomioon ajoneuvon käännökset negatiivisena vaikutuksena transitiotodennäköisyyteen. He demonstroivat sovitustarkkuuden parantumista 1–5 prosenttiyksiköllä riipuen liikeradan havaintoväleistä.

Koller et al. (2015) pyrkivät optimoimaan Newson ja Krummin HMM-algoritmin laskentanopeutta hylkäämällä matriisiesityksen ja Viterbi-algoritmin, ja sen sijaan mallintamalla pistekandidaatit verkkona, joka ratkaistaan kaksisuuntaisella Dijkstran algoritmilla (bidirectional-Dijkstra). Kaarien painot olivat aluksi tuntemattomat ja ne laskettiin vasta siinä vaiheessa, kun algoritmi päätyi solmuun, josta kaaret lähtevät. Lyhyimpien reittien haku tieverkosta transitiotodennäköisyyksien selvittämiseksi on tyypillisesti raskain operaatio, joka HMM-karttasovituksessa täytyy tehdä. Kandidaattipareja, joiden välille transitiotodennäköisyys täytyy laskea, on potentiaalisesti  $k^2(n-1)$ , jossa  $k$  on etsittävien segmenttiekandidaattien maksimilukumäärä ja  $n$  on liikeradan havaintopisteiden lukumäärä. Esimerkiksi korkeintaan kuuden kandidaatin haulla ja 500 havaintopisteen liikeradalla tämä tarkoittaa 17964 lyhyimmän reitin hakua. Kollerin et al. (2015) algoritmissa verkon kaarien painoiksi tulee yhdistelmä emissio- ja transitiotodennäköisyyksistä. Myös tässä algoritmissa Newsonin ja Krummin (2009) alkuperäinen absoluuttiseen etäisyyseroon perustuva transitiotodennäköisyys korvattiin suhteellisella erolla. Kirjoittajat demonstroivat artikkelissaan algoritmin pystyvän yhtä hyvään sovitustarkkuuteen kuin Newsonin ja Krummin (2009) algoritmi, mutta jopa 30–45 % pienemmällä määrällä lyhyimmän reitin hakua.

Yang ja Meng (2014) ja Xu et al. (2014) toteuttivat erittäin harvoille liikeradoille (60–300 s) karttasovitusalgoritmit, jotka perustuivat ehdollisiin satunnaiskenttiin (Conditional Random Field, CRF). Satunnaiskenttämalleissa on ideana se, että riippuvuudet mallinnetaan ketjun sijaan vapaammin verkkona. HMM:iin verrattuna CRF:n etuna on se, että tiukka aikariippuvuus edellisestä havainnosta poistuu mallista, joten yksittäisen tilan todennäköisyydessä voidaan ottaa huomioon kaikki menneet ja tulevat havainnot (Lafferty et al. 2001). CRF:iin perustuvat algoritmit noudattavatkin jotakuinkin samaa logiikkaa, kuin IVMM-algoritmi. On selvää, että satunnaiskentille suoritettavat laskuoperaatiot ovat paljon raskaampia kuin HMM:lle. Xu et al. (2014) pyrkivät hyödyntämään eri reittien suosiota liikeratojen karttasovituksessa, koska ihmiset käyttävät yleensä samoja reittejä samoihin vuorokaudenaikoihin. Täten liikeradan puutteellista dataa kompensoidaan *a priori* tiedolla tieverkon normaalista käytöstä. He saavuttivat noin 90 % sovitustarkkuuden 180 s havaintovälin datalla. Yang ja Meng (2014) käyttivät 16 eri muuttujaa taksien paikannus- ja telemetriadatan karttasovitukseen ja he saavuttivat noin 97–98 % sovitustarkkuuden 60–120 s havaintovälillä.



## 4.2 Digitaalisen tieverkon merkitys karttasovituksessa

Käytettävää tieverkkoa kuvaava digitaalinen kartta on tärkeä osa karttasovitusalgoritmeja. Digitaaliset tieverkot mallinnetaan usein topologista vektorimallia käyttäen. Muita vaihtoehtoja voivat olla esimerkiksi kantakartoilla esiintyvät tavat mallintaa tiealueet aluemaisina kohteina tai niiden reunaviivat viivamaisina kohteina, mutta tällöin ei pyritä mallintamaan tieverkkoa, vaan tiealueiden ulottuvuuksia esimerkiksi rakentamisen tai kaupunkisuunnittelun tarpeisiin. Karttasovituksen kannalta topologiset keskilinjaesitykset ovat yleensä huomattavasti käyttökelpoisempia.

Yleisin tapa kuvata tieverkkoa on mallintaa yksittäisten ajoratojen keskilinjat. Tällöin monikaistainen tie, jonka kaistat muodostavat yhtenäisen ajoradan, kuvataan vain yhdellä viivalla, joka kulkee ajoradan keskellä. Mikäli ajorata jaetaan fyysisesti esimerkiksi keskikorokkeella, kuvataan se toisissa aineistoissa tällöin kahdella viivalla ja joissakin edelleen vain yhdellä. On huomionarvoista, että tällaista kuvaustapaa käytettäessä tavanomainen kaksikaistainen maantie, jolla kulkee yksi kaista kumpaankin suuntaan, mallinnetaan siis yhdellä keskilinjaa kuvaavalla viivalla, jota voidaan kulkea kumpaan tahansa suuntaan. Esimerkiksi moottoritien ajoratojen keskilinoja kuvaavat viivat puolestaan ovat kuljettavissa vain yhteen suuntaan.

Digitaalisten tieverkkojen mallinnukseen sovelletaan yleensä topologisia sääntöjä. Nämä säännöt määrittelevät muun muassa tieverkon yhdistyvyyden, joka kuvaa todellisten reittien yhdistyvyyttä. Tällöin esimerkiksi moottoritien ylikulkusiltaa kuvaava viiva kulkee moottoritien keskilinjojen ylitse muodostamatta yhteisiä (eksplisiittisiä) taitepisteitä niiden kanssa. Tällaisten sääntöjen avulla on mahdollista kulkea ohjelmallisesti tieverkon läpi tai etsiä siitä erilaisia reittejä ilman että tehdään mahdottomia siirtymiä suhteessa fyysiseen todellisuuteen (esimerkiksi muodostamalla reitti suoraan moottoritien ylittävältä sillalta alas moottoritielle).

Julkisen hallinnon suositus JHS 188 (2014) kuvaa tällaisen topologisesti eheän ajoradan keskilinjan viivageometriaan perustuvan tieverkon mallinnussäännöt. Mallia suositellaan käytettäväksi digitaalisten tieverkkojen kuvaamiseen Suomessa. Malli perustuu linkeihin, jotka ovat suunnattuja topologisia yhteyksiä, jotka ovat pääsääntöisesti liittymävalin mittaisia ja linkejä toisiinsa yhdistäviin pistemäisiin solmuihin.

Topologisesti mallinnettua digitaalista tieverkkoa voidaan käsitellä myös graafina. Graafiteoriassa kuvataan entiteettejä solmuina ja niiden välisiä suhteita kaarina. Kokonaisuutta kutsutaan graafiksi tai verkoksi. Se on topologinen kuvaus pisteistä ja niiden välisistä suhteista. Graafina voidaan kuvata monia todellisia asioita, kuten tieverkkoa, mutta yhtä hyvin esimerkiksi jonkin laitteen sähkökaaviota tai ihmisten muodostamaa sosiaalista verkostoa. Puhtaasti graafiteorian mukaisena graafina esimerkiksi tieverkosta menetetään tieto metriksistä ominaisuuksista ja riippuvuuksista, kuten solmujen sijainnista, kaarien pituuksista tai solmujen etäisyyksistä toisistaan. Graafissa säilyy ainoastaan tieverkon topologia (Wilson 1996).

Graafit ovat joko suunnattuja tai suuntaamattomia. Suuntaamattomassa graafissa verkon kaaria voidaan kulkea kumpaan suuntaan tahansa, ja vastaavasti suunnatussa graafissa kaaren kulkusuunta on määritelty. Mikäli suunnatussa graafissa kahden solmun välillä on mahdollista kulkea kumpaankin suuntaan, täytyy se määritellä kahdella kaarella. Esimerkiksi

kulkeminen solmujen  $v_1$  ja  $v_2$  välillä määritellään tällöin kahdella kaarella  $e_{1,2}$  ja  $e_{2,1}$ . Koska yleensä käytännön sovellusten kannalta esimerkiksi katujen pituudet ovat merkittäviä, voidaan graafi kuvata painotettuna. Tällöin jokaisella kaarella on paino, mikä voi olla esimerkiksi katuosuuden pituus. Graafia voidaan analysoida erilaisilla algoritmeilla sovelluksen kannalta mielekkäällä tavalla, tai siitä voidaan esimerkiksi hakea eri tavoin optimoituja reittejä. Paino voi olla yksinkertaisen pituuden sijaan esimerkiksi matka-aika, mikä on saatu yhdistämällä nopeusrajoitustieto segmenttien pituuteen. Verkon kaarien painoihin voisi yhtä hyvin mallintaa esimerkiksi ruuhkatietoja tai tietullien aiheuttamia rahallisia kustannuksia. (Wilson 1996.)

#### 4.2.1 Paikkatiedon laatu

Karttasovituksen kannalta on relevanttia huomioida, että digitaaliseen karttaan liittyy useita eri virhelähteitä. Paikkatiedon laatua määritellään ISO 19157 standardissa. Paikkatiedon laatu jaetaan standardissa eri laatutekijöihin, joita ovat sijaintitarkkuus, täydellisyys, looginen eheys, temaattinen tarkkuus sekä ajallinen laatu. Sijaintitarkkuus tarkoittaa aineiston absoluuttista tai suhteellista sijaintitarkkuutta verrattuna tarkemmalla mittausmenetelmällä tuotettuun totuuteen. Absoluuttisella tarkkuudella tarkoitetaan kohteiden sijainnin eroa totuudesta ja suhteellisella tarkkuudella puolestaan esimerkiksi kohteiden välisten etäisyyksien eroa toisistaan verrattuna totuuteen. Aineiston täydellisyydellä tarkoitetaan puuttuvien tai ylimääräisten kohteiden esiintymistä aineistossa. Temaattisella tarkkuudella tarkoitetaan kohdetta kuvailevien tietojen paikkansapitävyyttä tai siihen liittyvien kvantitatiivisten arvojen tarkkuutta. Sekä täydellisyyttä että temaattista tarkkuutta voidaan mitata vertaamalla otantaa kerättyyn totuutena pidettävään tietoon, ja tutkimalla eri virhetyyppien suhteellista esiintyvyyttä otannassa. Loogisella eheydellä tarkoitetaan sitä, missä määrin aineisto noudattelee tietomallin loogisia sääntöjä. Loogiset säännöt liittyvät esimerkiksi siihen, onko aineisto yhdenmukaista sen tietomallin kanssa, ovatko kohteilla esiintyvät arvot määriteltujen arvojoukkojen mukaisia ja noudattaako aineiston topologia tietomallin määritysten mukaisia topologisia sääntöjä. Spesifit loogiset säännöt riippuvat sovellusalueesta, johon aineisto liittyy. Esimerkiksi tieverkon linkkien puutteellinen yhdistyvyys tai rakennuksen ja vesialueen päällekkäisyys voivat olla loogisen eheyden virheitä. Ajallisella laadulla tarkoitetaan aikaan liittyvien tietojen tarkkuutta ja eheyttä (esimerkiksi järjestystä) aineistossa ja sen lisäksi aineiston ajantasaisuutta. Näiden mitattavien laatutekijöiden lisäksi on olemassa kuvailevia laatutekijöitä, jotka koskevat esimerkiksi aineiston metatietojen paikkansapitävyyttä, historiatietoja ja käyttökokemuksia. (ISO 19157.)

Aineiston laatua arvioitaessa erotetaan toisistaan usein sisäinen (tuottajan) laatu ja ulkoinen (käyttäjän) laatu. Sisäinen laatu tarkoittaa aineiston laadukkuutta verrattuna valittuihin aineiston tuottamisen määrittämiin ja tietomalliin, johon tieto kerätään. Aineiston virheet ovat tällöin siis virheitä suhteessa spesifikaatioihin ja tietomalliin – ei sinänsä todellisuuteen. Ulkoinen laatu tarkoittaa puolestaan aineiston soveltuvuutta käyttäjän tiettyyn käyttötarkoitukseen. (ISO 19157.)

Digitaalisessa tieverkkoaineistossa esitetyn tien keskilinjan sijainnin erotus suhteessa todelliseen ajoradan keskilinjaan johtuu kartan luomisesta tapahtuneesta mittaamiseen, digitointiin tai karttaprojektioon liittyvistä virheistä. Toisaalta, jos GNSS-paikannuksella mitataan ajoneuvon sijaintia, jota verrataan ajoradan keskilinjaan, eroaa mitattu sijainti äärettömällä mittaustarkkuudellakin keskilinjan sijainnista, koska ajoneuvot kulkevat käytännössä aina

kaistoilla, joiden keskilinjat poikkeavat ajoradan keskilinjasta. Tämä virhe on sitä suurempi, mitä leveämmällä tiellä kuljetaan. Tämä on esimerkki siitä, että keskilinja-aineiston sijaintitarkkuutta voidaan pitää tuottajan laadun kannalta korkeampana, kuin karttasovitusta hyödyntävän sovelluksen ulkoisen laadun kannalta. Digitaalisessa tieverkossa voi olla myös kokonaan puuttuvia kohteita tai ylimääräisiä kohteita verrattuna fyysiseen todellisuuteen. Tieverkkoaineiston ajantasaisuus on usein puutteellinen etenkin alueilla, jotka kehittyvät. Mikäli karttasovituksessa hyödynnetään tieverkon topologiaa, aiheuttavat tieverkon topologia-virheet virheitä myös karttasovitukseen ja mikäli tieverkon attribuutteja hyödynnetään karttasovituksessa, kasautuvat niihin liittyvät virheet karttasovitustuloksen virheiksi. (Quddus 2006.)

Kirjallisuudessa toteutetuissa karttasovitusalgoritmeissa on hyödynnetty monia eri muuttujia, joita paikannusdatasta tai sovelluksen erityispiirteistä on johdettavissa. Monesti nämä menetelmät asettavat vaatimuksia myös tieverkon attribuuteille. Mikäli karttasovitusmenetelmässä esimerkiksi pyritään mallintamaan tien leveyden suhde havaintopisteen etäisyyteen tien keskilinjaesityksestä, täytyy tieverkon segmenteille olla määriteltynä leveystieto tai vähintään luokittelutieto, josta se on järkevästi johdettavissa. Sama pätee esimerkiksi ajoneuvon nopeustietojen vertaamiseen tieverkon nopeusrajoituksiin.

#### 4.2.2 Lähimmän naapurin haku

Karttasovituksen perustoiminnallisuuteen kuuluu etsiä tieverkosta sellaisia segmenttejä, jotka ovat lähellä havaintopistettä. Tieverkko tuleeikin mahdollisuuksien mukaan saada karttasovitusta varten sellaiseen formaattiin, josta voidaan tehdä tehokkaasti spatiaalisia hakuja. Käytännössä luodaan spatiaalinen indeksi, josta voidaan hakea tehokkaasti sellaiset tieverkon osat, joita halutaan kulloinkin käyttää. Yleensä tällaista lähimmän naapurin hakua rajoitetaan sekä etäisyyden että lähimpien kohteiden maksimilukumäärän mukaan.

Monissa karttasovitusalgoritmeissa tieverkon segmenttikandidaattien haku perustuu virhe-ellipsiin, joka voidaan rakentaa Kalman-suotimen tuloksena saatavista muuttujakohtaisista virhevariansseista tai monien GNSS-vastaanottimien laskemista varianssi-kovarianssimatriiseista. Virhe-ellipsi määritellään yleensä luottamusväleihin perustuen eli datan varianssin ja keskihajonnan perusteella. Keskihajonta  $\sigma$  vastaa 68 % luottamusväliä, mikä tarkoittaa sitä, että 68 % tuloksista osuu tälle välille. Yleensä sovelluksissa käytetään 95 % luottamusväliä, mikä vastaa noin kahta keskihajontaa ( $1.96$  tarkalleen). Zhao (1997) huomauttaa, että koska digitaalisessa kartassa on myös virheitä ja virheettömänäkin se on yleensä mallinnettu ajoradan keskilinjakuvausta käyttäen, tulisi karttasovitussovelluksissa käyttää vieläkin suurempaa luottamusväliä, esimerkiksi 99 % ( $3.03\sigma$ ). Jos luottamusväli muodostetaan havaitun ja todellisen sijainnin välisen etäisyyden virheestä, saadaan pyöreä virheympyrä. Yleensä kuitenkin sijaintivirhe jaetaan x- ja y-komponentteihinsa, jolloin saadaan virhe-ellipsi, olettaen että komponenttien virheet ovat erisuuruiset. Komponenttien keskinäinen korrelaatio puolestaan ratkaisee ellipsin orientaation. Käytännössä yleensä havaitaan virhe-ellipsin pääakselin suunnan seurailevan ajoneuvon kulkusuuntaa.

Virhe-ellipsin pääakselin suunta  $\beta$  voidaan ratkaista kaavalla

$$\beta = \frac{\pi}{2} - 0,5 \tan^{-1} \left( \frac{2\sigma_{xy}}{\sigma_x^2 - \sigma_y^2} \right) \quad (2)$$

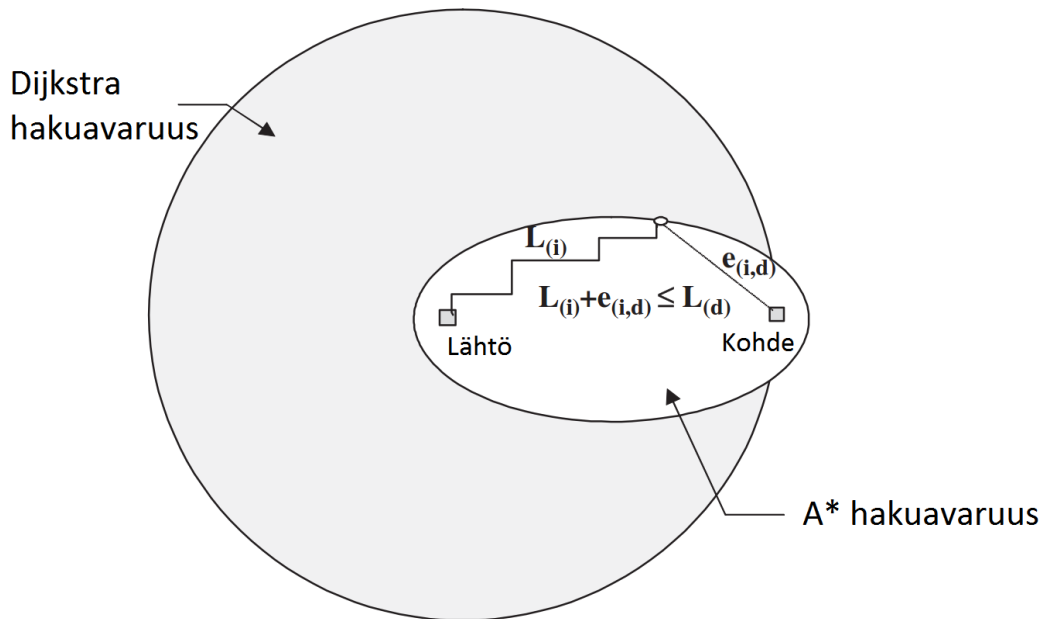
missä  $\sigma_x$  on x-akselin suuntaisen virheen varianssi,  
 $\sigma_y$  on y-akselin suuntaisen virheen varianssi ja  
 $\sigma_{xy}$  on näiden kovarianssi.

### 4.2.3 Lyhyimmän reitin ongelma

Yksi yleisimmistä liikennetelematiikan sovelluksista, joihin hyödynnetään graafeja, on lyhyimmän reitin haku tieverkosta, ja myös monissa karttasovitusalgoritmeissa hyödynnetään sitä. Käytetyin algoritmi ongelman ratkaisemiseksi on Dijkstran algoritmi. Siinä lähtöpisteeseen yhdistyvät solmut lisätään prioriteettijonorakenteeseen järjestämällä ne kokonaiskustannuksen perusteella nousevaan järjestykseen. Algoritmi siirtyy seuraavaan solmuun valitsemalla jonosta pienimmän kokonaiskustannuksen mukaisen uuden solmun. Aina kun algoritmi siirtyy uuteen solmuun, lisätään prioriteettijonoon sellaiset uudet solmut, joihin siitä voidaan kulkea ja joita jonossa ei vielä ole. Algoritmi etenee lyhintä mahdollista reitin kokonaispituutta tutkien niin pitkään, että kohdesolmu saavutetaan. (Dijkstra 1959.)

Voidaan helposti nähdä, että Dijkstran algoritmi tutkii verkkoa etenemällä lähtöpisteestä tasaisesti joka suuntaan olettaen, että verkko on yhtä tiheä kaikissa suunnissa. Tämä aiheuttaa intuitiivisesti ajatellen turhaa työtä, jos ajatellaan tavanomaista ajoneuvon reititystä kahden tieverkon pisteen välillä. Esimerkiksi Lahdesta Helsinkiin lyhintä reittiä haettaessa ei vaikuta järkevältä etsiä lyhyimpiä reittejä tasaisesti sekä Lahden etelä- että pohjoispuolelta, kun Helsinki sijaitsee etelässä.

Ongelmaa on yritetty ratkaista useilla heuristisilla lyhyimmän reitin algoritmeilla, joista A\* (A-star) on yleisin. A\* asettaa jokaiselle solmulle  $i$  prioriteetin käyttämällä vapaasti valittavaa heuristista funktiota  $F(i) = L(i) + e(i, d)$ , jossa  $L(i)$  on koko tähänastisen reitin kustannus kuten Dijkstran algoritmissakin, ja  $e(i, d)$  on heuristinen osa, eli arvioitu kustannus nykyisestä solmusta kohdesolmuun. Heuristinen funktio voi olla melkein mitä tahansa, mutta yleisesti tieverkossa reitittämisessä käytetään solmun todellisesta pistesijainnista laskettua euklidista etäisyyttä kohdesolmuun. Tällöin esimerkiksi Lahden pohjoispuolen solmut tulisivat tarkistetuksi vasta eteläpuolen solmujen jälkeen tai kenties ei lainkaan ennen kuin algoritmi jo päättyisi kohdesolmun löytymiseen. Dijkstran algoritmin hakuvaruutta voidaan ajatella tasaisesti laajenevana ympyränä, ja A\*-algoritmin hakuvaruutta ellipsinä, joka suuntautuu lähtösolmusta kohdesolmua kohti. Hakuvaruuksien eroja on havainnollistettu kuvassa 6. (Fu et al. 2005.)



Kuva 6: Dijkstra ja A\* hakuavaruuksien vertailu (Fu et al. 2005).

Kun kohdesolmu löydetään A\*-algoritmissa ja reitin kokonaiskustannus  $L(d)$  saadaan selville, sellaiset solmut, joiden kertynyt kustannus ja heuristinen osa on alle reitin kokonaiskustannuksen, tutkitaan vielä ennen algoritmin loppumista, eli  $L(i) + e(i,d) \leq L(d)$ . Tämä takaa sen, että mikäli heuristiikka  $e(i,d)$  ei yliarvioi lopullista kustannusta, löydetään aina paras mahdollinen reitti. Heuristiikkaa joka ei yliarvioi kustannusta kutsutaan luvaloiseksi (engl. admissible). Jos käytetään ei-luvalloista heuristiikkaa, on mahdollista toisaalta saavuttaa huomattavia hyötyjä algoritmin nopeuteen, mutta ei ole taattua, että optimaalinen reitti löydetään. (Fu et al. 2005.)

Algoritmien kompleksisuutta eli suoritukseen vaadittavaa aikaa tai niiden varaaman muistin määrää kuvataan yleensä ordo-notaatioissa, eli niin sanotussa Big O-notaatioissa. O-notaatioissa kuvataan algoritmin suorituskyyvyn asympotoottista ylärajaa eli funktion  $O(g(n))$  kasvua, kun syötteen koko  $n$  kasvaa, kun syöte on jotakin alarajaa suurempi. Se kuvaa siis algoritmien suhteellista suorituskyykyä syötteen kokoon nähden, kun vakiot, vakiokertoimet ja muut kuin nopeimmin kasvava termi jätetään pois (MIT). Dijkstran algoritmin aikakompleksisuus pinorakennetta käyttäen on  $O(E + V \log(V))$ , jossa  $E$  on graafin kaarien lukumäärä ja  $V$  solmujen lukumäärä (Berg et al. 2008). A\*-algoritmin suoritusaajan yläraja on eksponentiaalinen suhteessa oikean reitin pituuteen, mikä seuraa huonon heuristiikan käyttämisestä. Toisaalta optimaalisella heuristiikalla suoritusaika on lineaarinen suhteessa oikean reitin pituuteen, eli heuristiikka ratkaisee suoritusaajan, eikä O-notaation käyttö ole täten yleisellä tasolla hyödyllistä A\*-algoritmia arvioitaessa (Russell ja Norvig 2010).

### 4.3 Yhteenveto

Algoritmeista kirjoitettujen artikkeleiden aikajanaa tutkimalla voidaan todeta, että varhaiset, ennen GPS-järjestelmää kirjoitetut artikkelit yrittivät parantaa pelkillä DR-sensoreilla toteutetun paikannuksen tarkkuutta karttasovituksen avulla yksinkertaisilla geometrisilla menetelmillä. GPS-järjestelmän perustamisen jälkeen 1990-luvulla heikotarkkuuksista GPS-paikannusta yritettiin edelleen parantaa monen eri sensorin datan yhdistämisellä Kalman-suotimen ja karttadatan avulla. Karttasovitusta tehtiin edelleen lähinnä geometrisin menetelmin. Kun SA-heikkennys poistettiin GPS-järjestelmästä, tutkimus keskittyi karttasovituksen tarkkuuden parantamiseen algoritmeilla, jotka ottavat tieverkon topologian huomioon paremmin. Tätä seurasivat muutamat kokeilut Dempster-Shaferin uskottavuusteoriolla, jota seurasi suuri joukko sumeaa logiikkaa hyödyntäviä menetelmiä, joiden tarkkuus oli jo melko korkea. 2010-luvulla Newsonin ja Krummin (2009) esimerkin mukaisesti eniten artikkeleita kirjoitettiin HMM-menetelmistä ja niitä voikin pitää viimeisimpänä ja parhaana ratkaisuna ongelmaan. Myös muutamat muut erilaiset menetelmät, jotka kaikki kuitenkin hyödyntävät lyhyimpien reittien esilaskentaa tieverkosta, kuten HMM-algoritmikin, ovat osoittaneet pystyvänsä karttasovitukseen korkealla tarkkuudella (esimerkiksi ST-Matching, IVMM, Quddus ja Ochieng (2014) topologinen algoritmi). Kirjallisuuden perusteella vaikuttaakin siltä, että avain onnistuneeseen karttasovitusongelman ratkaisuun on liikeradan pisteiden ominaisuuksien lisäksi niiden välisten siirtymien mallintamisessa mahdollisimman hyvin, johon pistekandidaattien välisten lyhyimpien reittien ominaisuudet soveltuvat hyvin.

Kirjallisuudessa on pyritty löytämään yhä kehittyneempiä menetelmiä haasteellisemmän datan karttasovitukseen, mutta tämä haastavuus on käytännössä aina seurannut suuresta havaintovälistä tai toisinaan sijainninmäärityksen epävarmuudesta. Yksikään tutkittu artikkeli ei yrittänyt suoraan ratkaista ongelmaa, jonka haastavuus seuraisi erittäin tiheästä ja heterogeenisestä tieverkosta, kuten tässä tutkimuksessa pyritään tekemään.

Kirjallisuuden perusteella tavallinen absoluuttista paikannusta hyödyntävä GNSS-vastaanotin, joka kykenee alle 10 metrin paikannustarkkuuteen riittää korkean karttasovitustarkkuuden saavuttamiseen tavanomaisissa sovelluksissa, mikäli käytetään kehittyneitä algoritmeja, esimerkiksi kuten Newson ja Krumm (2009), Lou et al. (2009) tai Xu et al. (2015). Vastakkaisen näkemyksen esittävät Lakakis et al. (2004), jotka suosittelevat DGPS-paikannuksen käyttämistä kalustonhallintasovelluksissa sovitustarkkuuden parantamiseksi – toisaalta heidän tutkimuksessaan käytetty sovitusmenetelmä ei ollut yhtä kehittynyt kuin uudemmat algoritmit. Paikannusdatan frekvenssiä voidaan pitää yhtä tärkeänä ominaisuutena kuin paikannuksen sijaintitarkkuutta. Käytettävän digitaalisen tieverkon kartan mittakaava ja laatu ovat myös ratkaisevassa asemassa karttasovituksen onnistumisen kannalta.

Karttasovitusalgoritmien suorituskykyä mitataan yleensä tutkimalla liikeradan sovitettujen pisteiden sijoittumista oikealle tieverkon segmentille vertaamalla karttasovituksen tulosta tunnettuun referenssiaineistoon. Eri algoritmien suorituskyvyn arviointi raportoitujen onnistumisprosenttien perusteella on kuitenkin hyvin haastavaa, sillä suorituskykyyn vaikuttaa itse algoritmin perusteella ainakin käytettävä paikannusteknologia, digitaalinen kartta sekä testialueen erityisominaisuudet. Jos liikerataa pystytään mittaamaan tiheästi, esimerkiksi 1–5 s välein, karttasovitusongelma on paljon helpompi ja sen toteuttamiseen voidaan soveltaa erilaisia menetelmiä kuin esimerkiksi harvalla 30–120 s havaintovälillä. Maantieympäristössä yksinkertaisimmankin algoritmin onnistumisprosentti on todella korkea, kun taas kaupunkiolosuhteissa ongelmasta tulee huomattavasti haastavampi. Mitä tiheämpi katuverkko

on, sitä vaikeampi ongelmasta tulee algoritmille. Esimerkiksi Srinivasan et al. (2003) raportoi DST:hen perustuvalla karttasovitusalgoritmille 98,5 % sovitustarkkuuden, mutta Quddus et al. (2006) saavutti sitä käyttäen samankaltaisella paikannusdatalla, mutta tiheämmässä tieverkossa vain 80,2 % sovitustarkkuuden. Tutkimuksia, joissa olisi tutkittu kattavasti suurta osaa erilaisista algoritmeista, ei ole. Luotettavia tuloksia algoritmien keskinäisestä sovitustarkkuudesta voidaan saada vain sellaisista tutkimuksista, joissa on verrattu kattavissa kenttätesteissä algoritmeja käyttäen samaa havaintodataa, tieverkkoa ja referenssiaineistoa.

Tämän tutkimuksen sovelluksessa haluttiin selvittää ajoneuvojen kulkemia reittejä toisin kuin suuri osa kirjallisuudessa esiintyvistä sovelluksista, jotka pyrkivät ensisijaisesti joko selvittämään tieverkon segmentin, jolla yksittäinen havaintopiste sijaitsee tai määrittämään ajoneuvon sijainnin tarkemmin. Tämä vaatimus puoltaa sellaisten globaalien algoritmien käyttämistä, jotka pystyvät löytämään todennäköisimmät reitit tieverkkoa pitkin sen sijaan, että haettaisiin vain todennäköisimpiä sijainteja havaintopisteille.

Newsonin ja Krummin (2009) HMM:n tyylliseen algoritmiin pohjautuvilla menetelmillä on kirjallisuudessa saavutettu korkeita sovitustarkkuuksia jopa melko harvaa dataa käyttämällä. Lisäksi menetelmä sovittaa havaitun liikeradan kokonaiseksi todennäköisimmäksi tieverkkoa pitkin kulkeväksi reitiksi, mikä on tavoitteena myös tässä tutkimuksessa. Globaalina algoritmina, jonka tietorakenteet kasvavat tavanomaisilla syötteillä verrattain suuriksi, sen laskennallinen suorituskyky ei lähtökohtaisesti sovellu reaaliaikaista karttasovitusta vaativiin sovelluksiin. Tässä tutkimuksessa sovitustarkkuudella on kuitenkin paljon suurempi painoarvo, kuin laskennallisella suorituskyvyllä, joten menetelmä on myös siltä osin sopiva. Kirjallisuuskatsauksen perusteella tässä tutkimuksessa päätettiin toteuttaa tällainen HMM-algoritmi ja testata sen soveltuvuutta Vantaan kaupungin katujen kunnossapidon reittien karttasovitukseen.

Toisena algoritmivaihtoehtona päätettiin toteuttaa sumeaa logiikkaan pohjautuva karttasovitusalgoritmi. Sumeaa logiikkaa käyttämällä on mahdollista mallintaa mielivaltaisen monia parametreja ja määritellä niiden vaikutukset intuitiivisesti ja helposti ymmärrettävästi selkeään viitekehykseen, mikä tekee menetelmästä houkuttelevan vaihtoehdon. Sumeaan logiikkaan perustuvia menetelmiä ei juuri löydy kirjallisuudesta 2010-luvulta eteenpäin, eikä kirjallisuudesta löydy yhtäkään sumeaa logiikkaa hyödyntävää menetelmää, joka noudattelisi samankaltaisia tieverkossa siirtymisen mallintamisen periaatteita, jotka Lou et al. (2009) ja Newson ja Krumm (2009) esittelivät artikkeleissaan. Tutkimuksessa haluttiin selvittää, voiko sumeaa logiikkaa nojautuvia karttasovitusalgoritmeja kehittää hyödyntämällä edellä mainittujen tutkimusten käyttämiä menetelmiä, parantaen näin sovitustarkkuutta harvalle datalle ja tuottaen lopputuloksena karttasovitettuja reittejä pisteiden sijaan.

Seuraavassa luvussa esitellään toteutettavien algoritmien taustalla vaikuttavien menetelmien, eli Markovin piilomallien ja sumean logiikan teoriaa. Luvussa 6 esitellään tutkittava data ja analysoidaan sitä. Luvussa 7 esitellään algoritmien toteutus ja testaaminen.

## 5 Toteutettavien algoritmien teoriaa

Tutkimuksessa toteutettiin kaksi erilaista karttasovitusalgoritmia, jotka hyödyntävät erilaisia matemaattisia konsepteja. Ensimmäinen algoritmi hyödyntää Markovin piilomallia ja toinen algoritmi hyödyntää sumeaa logiikkaa. Seuraavissa luvuissa kuvataan lyhyesti näiden eri menetelmien teoriaa.

### 5.1 Markov-mallit

Markov-malleja käytetään temporaalisen tai muun sekventiaalisen datan tilastolliseen mallintamiseen. Niiden ajatuksena on se, että tulevaisuus on riippumaton menneisyydestä, jos nykyhetki tunnetaan. Markov-malleja voidaan siis käyttää tilastolliseen satunnaismuuttujien mallintamiseen, jos kuvattava ilmiö sopii jollain tasolla tällaiseen ajatukseen. Suurin hyöty prosessien mallintamisessa tällä tavalla on se, että niistä tulee mahdollisia ratkaista laskennallisesti tietokoneella eli niin sanottuja P-ongelmia, jotka on mahdollista ratkaista polynomisessa ajassa (verrattuna NP-ongelmiin, joita ei ratkaista polynomisessa ajassa, vaan esimerkiksi eksponentiaalisessa).

Yksinkertaisin Markov-malli on Markovin ketju. Satunnaismuuttujien jono on Markovin ketju, jos se toteuttaa Markov-ominaisuuden  $p(x_t|x_1, \dots, x_{t-1}) = p(x_t|x_{t-1})$ . Toisin sanoen jono on aikariippumaton, mikä tarkoittaa sitä, että yksittäinen satunnaismuuttuja ei riipu muista kuin sitä edeltävästä muuttujasta. Edellä kuvattu malli on ensimmäisen asteen Markovin ketju, sillä tilat riippuvat vain edeltävästä tilasta. Toisen asteen Markovin ketjussa satunnaismuuttujat riippuvat kahdesta edellisestä tilasta ja niin edelleen. Yksinkertaisimmissa malleissa sekä aika että tila-avaruus ovat diskreettejä, mutta voitaisiin myös käyttää malleja, joissa jompikumpi tai molemmat ovat jatkuvia, jollakin funktiolla kuvattavia muuttujia.

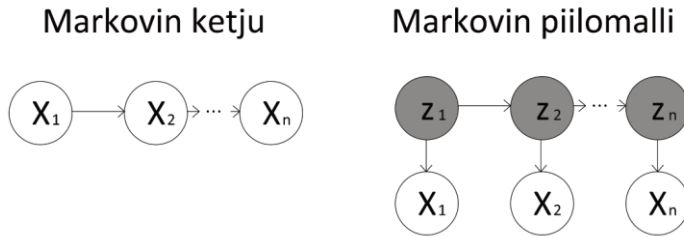
Markov-ketjuun kuuluu tilojen joukko  $Z$  ja transitiomatriisi  $T$ . Esimerkiksi päivän säätä voitaisiin mallintaa tilojen joukolla  $Z = \{\text{sateinen}, \text{pilvinen}, \text{aurinkoinen}\}$ . Päivät ovat diskreettejä päivämääriä, joista muodostuu tilojen sekvenssi  $x_1, x_2, \dots, x_n$ . Transitiomatriisi on  $Z \times Z$  kokoinen neliömatriisi joka kuvaa siirtymien todennäköisyyksiä tilojen välillä. Säätilojen transitiomatriisi voisi olla siis esimerkiksi

$$T = \begin{bmatrix} 0,6 & 0,3 & 0,1 \\ 0,2 & 0,6 & 0,2 \\ 0,1 & 0,4 & 0,5 \end{bmatrix} \quad (3)$$

missä ensimmäisellä rivillä on tilasiirtymien todennäköisyydet sateisesta säästä sateiseen, pilviseen ja aurinkoiseen vastaavasti. Seuraavalla rivillä taas on pilvisen tilan siirtymät muihin tiloihin ja niin edelleen. Transitiomatriisin rivien tulee summautua yhteen. Lisäksi malliin tarvitaan alkutodennäköisyydet  $s_0$ , jossa määritellään todennäköisyys aloittaa tietystä tilasta. Tällaista yksinkertaista mallia voitaisiin käyttää esimerkiksi vastaamaan kysymyseen: millä todennäköisyydellä malli pysyy nykyisessä tilassaan seuraavat  $n$  päivää? Tai: millä todennäköisyydellä mallin mukaan seuraavien 7 päivän sää muodostuu tiettyjen tilojen ketjusta (vaikkapa tila *aurinkoinen* kaikkina päivinä)? (Rabiner 1989.)



Markovin piilomalli (Hidden Markov Model, HMM) eroaa edellisestä siinä, ettei sen tiloja voida havainnoida suoraan. Sen sijaan havainnoidaan jotakin muuttujaa, joka liittyy tiloihin. Nämä muuttujat kuvataan havainnointi- tai emissiomatriisissa  $\varepsilon$  todennäköisyysjakaumana. Tällainen malli sopii erittäin hyvin kuvaamaan useita todellisen maailman temporaalisia tai sekventiaalisia ilmiöitä, joita voidaan mitata. Kuvassa 7 on havainnollistettu Markov-ketjun ja HMM:n rakennetta ja eroja.



Kuva 7: Markov ketjun ja Markovin piilomallin rakenteet.  $X$  on Markovin ketjussa havaittu tilojen sekvenssi. Markovin piilomallissa  $X$  kuvaa havaittua emissioiden sekvenssiä ja  $z$  piilotettuja tiloja.

HMM:n voi kirjoittaa muotoon

$$p(x_1, \dots, x_n, z_1, \dots, z_n) = s_0(z_1) \varepsilon_{z_1}(x_1) \sum_{k=2}^n T(z_{k-1}, z_k) \varepsilon_{z_k}(x_k) \quad (4)$$

missä  $s_0$  on alkutodennäköisyysjakauma eri tiloille,

$T$  on transitiomatriisi,

$\varepsilon$  on emissiomatriisi,

$z$  on tila,

$n$  on havaintojen lukumäärä ja

$x$  on yksittäinen havainto.

Kaksi tärkeää algoritmia, joita käytetään HMM:n todennäköisyysjakaumien laskentaan, ovat Forward-backward -algoritmi ja Viterbi-algoritmi. Forward-backward -algoritmia voidaan käyttää kaikkien tilojen posterioritodennäköisyysjakaumien laskemiseen mallista havaitulla sekvenssillä, ja Viterbi-algoritmilla voidaan etsiä yksittäinen todennäköisin tilojen polku mallista havaitulla sekvenssillä. Molemmat algoritmit hyödyntävät niin sanotun dynaamisen ohjelmoinnin ideaa ja mallin Markov-ominaisuutta, eli aikariippumattomuutta, todennäköisyyksien laskemiseen polynomisessa ajassa (Jurafsky ja Martin 2014).

Forward-backward-algoritmin aikakompleksisuus on  $O(S^2N)$ , jossa  $S$  on tilojen lukumäärä ja  $N$  on havainnoidun sekvenssin pituus. Viterbi-algoritmin aikakompleksisuus puolestaan on  $O(S^2T)$  jossa  $S$  on tilojen lukumäärä ja  $T$  on mahdollisten tilasiirtymien lukumäärä.

Markov-malleja on kehitetty 1900-luvun alusta lähtien, ensimmäiseksi venäläisen Andrey Markovin toimesta. Markovin ketjua hyödynnetään monilla aloilla kuvaamaan satunnaisia tapahtumaketjuja muun muassa bioinformatiikassa, taloustieteessä ja kemiassa. Esimerkiksi Googlen PageRank-algoritmi pohjautuu Markov-ketjuun (Langville ja Meyer 2006). Yksi paljon käytetty menetelmä on Monte Carlo Markov-ketju, jota hyödynnetään tilastotieteessä ja sitä soveltavilla aloilla approksimoimaan simuloimalla todennäköisyysjakaumia, joiden parametrit olisi muuten hyvin hankalaa saada selville. Markovin piilomalleja on kehitetty ja

käytetty 1960-luvulta lähtien. Ne ovat tulleet tärkeimmäksi menetelmäksi muun muassa puheentunnistuksessa (Manning ja Schütze 1999). Geoinformatiikassa Markov-malleja on hyödynnetty esimerkiksi spatiaalisten ilmiöiden simuloinnissa niin kutsuttujen spatiaalisten autoregressiivisten mallien yhteydessä (Zhang ja Goodchild 2002). Niitä käytetään ekstensiivisesti myös bioinformatiikassa muun muassa DNA-sekvenssin osien tunnistamiseen (Durbin et al. 1998).

## 5.2 Sumea logiikka

Lofti A. Zadeh kehitti 1960-luvulla sumean logiikan perusteet epätasällisyyden mallintamiseksi. Sumeassa logiikassa lingvistisiä ja epätasällisiä käsitteitä voidaan esittää matemaattisesti hyödyntämällä sumeita joukkoja. Esimerkiksi joukko *pitkät miehet* on konseptuaalisesti epätasällinen, eikä tavanomainen kaksiarvoinen Boolean logiikka sovi kovin hyvin kuvaamaan esimerkiksi 180-senttisen miehen kuulumista tähän joukkoon. Sumeissa joukoissa jäsenyysasteet ovat jatkuvia reaalilukuja arvovälillä  $[0.0, 1.0]$ , ja esimerkin tapauksessa pituuden pienentyessä jäsenyysaste joukossa pienenee ja vastaavasti pituuden kasvessa jäsenyysaste kasvaa. Boolean logiikalla 180-senttinen mies joko on tai ei ole pitkä, ja vastaavasti joko on tai ei ole lyhyt, luokkarajoista riippuen, mutta sumealla logiikalla jäsenyysaste luokassa *lyhyet miehet* voi olla vaikkapa 0,4 ja luokassa *pitkät miehet* 0,6. (Zadeh 1965.)

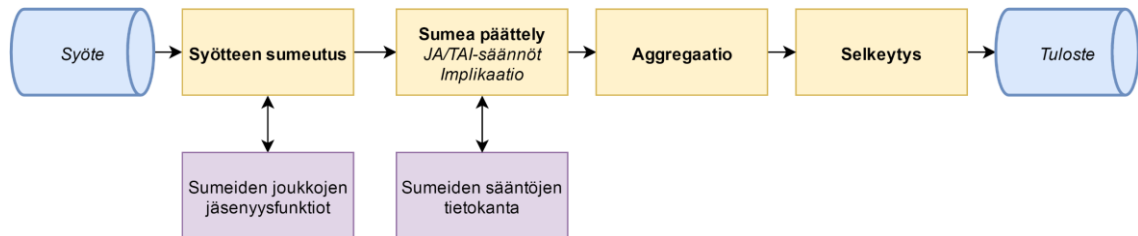
Sumea logiikka sopii hyvin heterogeenisten muuttujien yhdistämiseen ja normalisointiin samaan malliin, riippumatta siitä, ovatko ne numeerisia tai lingvistisiä. Esimerkki tästä on asiantuntijatiedon hyödyntäminen matemaattisessa mallissa. Se sopii monimutkaisten, epälineaaristen, matemaattisesti vaikeiden tai mahdottomien ongelmien approksimointiin siten, että saavutetaan riittävän hyvä lopputulos sovelluksen kannalta. Lisäksi se on tolerantti epätarkalle datalle (Zadeh 1973). Sumeaa logiikkaa käytetään usein säätötekniikassa ja erilaisissa teollisuuden sovelluksissa, esimerkiksi autoteollisuudessa. Sumeaa logiikkaa on käytetty myös geoinformatiikassa epätasällisten tai monitulkintaisten kohteiden mallintamiseen. Esimerkiksi maaperän eri maalajien luokitteluun perinteiset selvärajaiset menetelmät sopivat heikosti, sillä maalajit muodostavat transitiovyöhykkeitä, joissa maalaji vaihtuu toiseksi vähitellen (McBratney ja Odeh 1997).

On huomionarvoista, että sumean logiikan sumeus tai epätasällisuus ei synny teoriasta itsestään, vaan se kuvaa ennemminkin käytettävän datan mahdollista epätarkkuutta ja tapaa, jolla todellisen maailman ongelmat pystytään sitä hyödyntäen mallintamaan ja ratkaisemaan. Sumeat menetelmät voidaan säätää joustavasti niin tarkoiksi kuin kuvattava ongelma ja käytettävissä oleva data mahdollistaa ja vaatii. (Zadeh 1973.)

Pääsääntöisesti käytetään kahdenlaisia sumeita järjestelmiä (Fuzzy Inference System, FIS). Nämä ovat Mamdani-tyypin FIS ja Sugeno-tyypin FIS. Seuraavissa kappaleissa kuvataan perusteet molemmista.

### 5.2.1 Mamdani-tyyppin sumeat järjestelmät

Mamdani-tyyppin sumeassa järjestelmässä syötearvot sumeutetaan ja niille suoritetaan sääntöpohjainen sumea päättely, jonka tuloksena saadaan jäsenyysarvo sumeassa tulosjoukossa. Käytännön sovelluksissa sumea tulosjoukko yleensä vielä selkeytetään eli muutetaan takaisin diskreetiksi arvoksi. Kuvassa 8 havainnollistetaan Mamdani-tyyppin sumean järjestelmän osia ja toimintaa.



Kuva 8: Mamdani-tyyppin sumean järjestelmän toiminta (Quddus 2006 mukaillen).

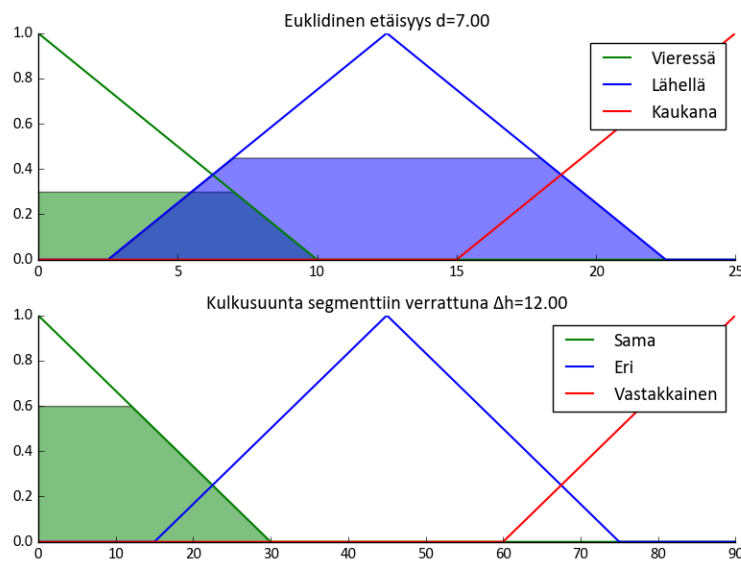
Sumean järjestelmän evaluoinnissa tapahtuu seuraavat viisi vaihetta:

1. *Syötearvojen sumeutus* tarkoittaa numeeristen syötteiden muuntamista sumeiden joukkojen jäsenyysasteiksi. Tämä tapahtuu jäsenyysfunktioiden avulla, jotka määrittelevät syötearvon sumean jäsenyyden määrät kaikissa sumeissa joukoissa koko mahdolliselle syöteavaruudelle. Jäsenyysfunktioiden muodon määrittäminen on yksi sumean järjestelmän suunnittelun tärkeimpiä vaiheita. Erilaisia yleisesti käytettyjä muotoja ovat esimerkiksi Gaussin käyrä, kolmiokäyrä, z-käyrä, s-käyrä, sigmoidi ja puolisuunnikas.
2. *Sumeiden sääntöjen evaluoinnissa* sumeat jäsenyysarvot muunnetaan sumeiden tulosjoukkojen jäsenyyksiksi. Tämä tapahtuu yhdistelemällä sääntöjä yksinkertaisin JA, TAI säännöin. Matemaattisesti tämä tapahtuu yleensä yhdistämällä kaksi tai useampia sumeita joukkoja laskemalla niiden maksimi tai summa (TAI-säännöt) tai vaihtoehtoisesti minimi tai tulo (JA-säännöt).
3. *Sumean tulosjoukon implikaatio* tarkoittaa sitä, millä menetelmällä sumean tulosjoukon jäsenyysaste määritellään yksittäiselle säännölle. Siinä käytetään samoja menetelmiä kuin JA-säännöissä, eli yleensä minimi tai tulo. Tässä vaiheessa saadaan jokaista sääntöä vastaavat sumeat tulosjoukot.
4. *Sumeiden tulosjoukkojen aggregaatiossa* kaikki tulosjoukot yhdistetään yhdeksi sumeaksi joukoksi, mikä vastaa lopullista sumeaa tulosta. Tässä vaiheessa käytetään samoja menetelmiä kuin TAI-säännöissä eli joukkojen maksimi tai summa.
5. *Sumean tulosjoukon selkeytyksessä* tulosjoukkojen aggregaatiosta saadaan tulokseksi yksittäinen arvo. Selkeytys tapahtuu yleensä jollakin seuraavista menetelmistä: massakeskipiste (centroid), puolittaja (bisector) tai valitsemalla korkeimman tulosjäsenyysfunktion pienin, keskimmäinen tai suurin kohta.

(MathWorks 2016.)

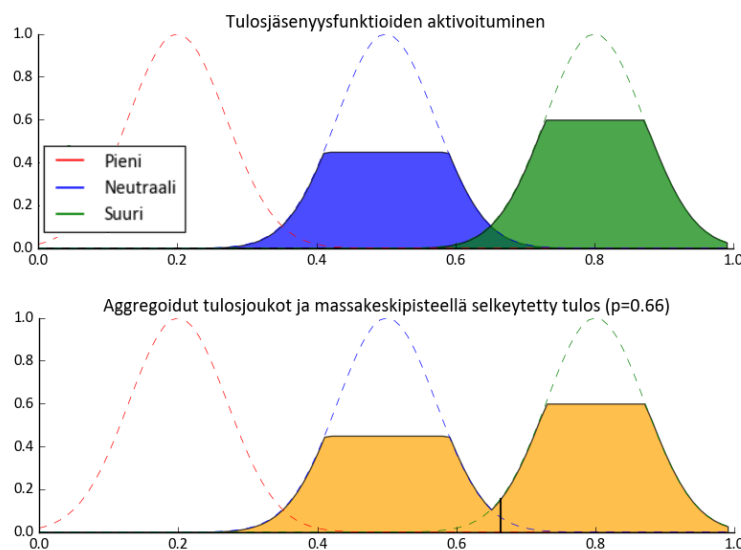
Kuvassa 9 on esitetty yksinkertaisen sumean järjestelmän syötteiden sumeutus. Järjestelmässä on kaksi syötettä: havaintopisteen etäisyys tiesegmenteistä  $d$  ja ajoneuvon kulkusuunnan ja segmentin orientaation välinen ero  $\Delta h$ . Kohdassa 1 esitetään kolmiomaiset jäsenyysfunktiot etäisyyden sumeille joukoille  $\{vieressä, lähellä, kaukana\}$  ja kohdassa 2 kulkusuunnan sumeille joukoille  $\{sama, eri, vastakkainen\}$ . Järjestelmässä on yksi tulosjoukko sovituksen todennäköisyydelle  $p$ , jonka jäsenyysfunktiot joukoille  $\{matala, neutraali, suuri\}$ , ovat Gaussin käyrän muotoisia. Järjestelmän säännöt ovat:

1. **JOS**  $d = \text{vieressä}$  **TAI**  $\Delta h = \text{sama}$  **NIIN**  $p = \text{suuri}$
2. **JOS**  $d = \text{lähellä}$  **TAI**  $\Delta h = \text{eri}$  **NIIN**  $p = \text{neutraali}$
3. **JOS**  $d = \text{kaukana}$  **TAI**  $\Delta h = \text{vastakkainen}$  **NIIN**  $p = \text{pieni}$



Kuva 9: Sumeat syötejoukot ja niiden aktivoituminen syötteellä 7.0, 12.0.

Kuvan 10 yläosassa esitetään tulosjäsenyysfunktioiden aktivoituminen syötteen sumeutuksen ja sääntöjen evaluoinnin jälkeen. Alaosassa esitetään aggregoidut tulosjoukot ja massakeskipistemenetelmällä selkeytetty tulos.



Kuva 10: Sumeat tulosjoukot sekä niiden aggregaatio ja selkeytetty tulos.

### 5.2.2 Sugeno-tyypin sumeat järjestelmät

Sugeno-tyypin sumeat järjestelmät eroavat Mamdani-järjestelmistä ainoastaan siinä suhteessa, että tulosjäsenyysfunktioiden sijaan niiden tulokset ovat syötearvojen funktioita. Jos Mamdani-tyypin järjestelmän sumea sääntö on siis esimerkiksi

$$JOS\ x = i\ JA\ y = j,\ NIIN\ p = z,$$

jossa  $z$  on sumea tulosjoukko, jolle on määritelty oma jäsenyysfunktionsa, niin Sugeno-järjestelmässä sääntö on

$$JOS\ x = i\ JA\ y = j,\ NIIN\ p = g(x, y),$$

jossa  $g$  on jokin syötearvojen funktio, esimerkiksi  $ax + by + c$ . Tämän kaltaista järjestelmää kutsutaan ensimmäisen asteen Sugeno-järjestelmäksi. Funktion sijaan tuloksena voi olla vakio, esimerkiksi  $p = 100$ . Tällöin kyseessä on nollannen asteen Sugeno-järjestelmä. (Takagi ja Sugeno 1985, MathWorks 2016.)

Jokaisen säännön antama tulos selviää yhdistämällä syötteiden saamat jäsenyysfunktioiden arvot samoilla JA/TAI-menetelmillä kuin Mamdani-järjestelmässäkin, ja säännöt yhdistetään yhdeksi tulokseksi niiden painotetulla keskiarvolla.

Sugeno-tyypin järjestelmien yhtenä etuna on niiden yksinkertaisempi matemaattinen optimointi, kuin Mamdani-tyypin järjestelmien. Niiden optimoimiseksi on kehitetty muun muassa neuroverkkoon perustuvia menetelmiä. Neuroverkko on matemaattinen optimointimenetelmä, jossa verkko mukautuu siihen syötettyyn dataan joko ohjatusti tai ohjaamattomasti neuroverkon tyypistä riippuen (Bishop 1995). Sugeno-tyypin sumean järjestelmän jäsenyysfunktioiden muotoja ja sääntöjä voidaan optimoida hyödyntämällä neuroverkkoa, joka optimoi nämä parametrit datavetoisesti. Menetelmän käyttö edellyttää oikean opetusdatan keräämistä ja syöttämistä neuroverkolle, mikä saattaa sovelluksesta riippuen olla helppoa tai vaikeaa valmistella. Sumeita järjestelmiä, jotka on optimoitu käyttäen neuroverkkoa, kutsutaan adaptiivisiksi neuro-sumeiksi järjestelmiksi (Adaptive Neuro Fuzzy Inference System, ANFIS). Työkaluja, joilla tämän kaltaisia järjestelmiä voi luoda on saatavissa ainakin kaupallisessa Matlab-ohjelmassa ja avoimen lähdekoodin R-tilastolaskentaohjelman kirjastona (MathWorks 2016, R Archive Network 2015).

## 6 Datan kerääminen

### 6.1 Järjestelmän kuvaus

Vantaan kaupungin ajoneuvot tallentavat liikkueensa sijaintihavaintoja ja lähettävät ne PPCT Finlandin keskitetylle palvelimelle. Tältä palvelimelta voidaan noutaa dataa ohjelmallisesti rajapinnan kautta. Rajapinta on RPC-tyylinen (Remote Procedure Call) XML-representaatioon nojaava Web-palvelu. Rajapinnasta on mahdollista noutaa dataa määrittelemällä aikaväli, jolta havaintoja haetaan ja rajaamalla tulos ajoneuvo- tai ajoneuvoryhmäkohtaisesti. Palvelua käytetään lähettämällä http-protokollan POST-pyyntö rajapinnan URI-osoitteeseen (Uniform Resource Identifier). POST-pyyntöön body-elementtiin upotetaan XML-dokumentti, joka kuvailee pyynnön. Vastaukseksi palvelu lähettää toisen XML-dokumentin, joka sisältää vastauksena kysytyjen ajoneuvojen havaintopisteet attribuutteineen halutulta aikaväliltä. Vastaus sisältää muun muassa ajoneuvon tunnisteen, koordinaatit maantieteellisessä WGS84-koordinaattijärjestelmässä, kulkusuunnan, nopeuden, aikaleiman ja ajoneuvon kanavatietojen muutokset.

Datan analysoimiseksi ja esittämiseksi tämä data kerätään Vantaan kaupungin omaan paikatietojärjestelmään. Datan tallentamiseen käytettävä tietokanta on PostgreSQL, joka on avoimen lähdekoodin relaatiotietokanta. PostgreSQL:ää kehittää sekalainen käyttäjien ja yritysten yhteisö, kuten monia muitakin avoimen lähdekoodin ohjelmistoja. Tietokannassa on spatiaalinen laajennus PostGIS, joka lisää tietokantaan tuen spatiaalisille tietotyypeille, -indekseille ja -funktioille (PostgreSQL Global Development Group 2016).

Edellä mainitun pyynnön suorittamisen ja tiedon tallentamisen automatisoinniksi suoritetaan prosessi Safe Softwaren FME:llä (Feature Manipulation Engine). FME on spatiaalisen datan käsittelyyn tarkoitettu ETL-ohjelma (Extract Transform Load), jolla on mahdollista rakentaa työkulkuja ja automatisoida tai ajastaa niitä. Se tukee satoja erilaisia dataformaatteja, sekä spatiaalisia että muita, ja sen työkulut perustuvat lukijoihin (reader) ja kirjoittimiin (writer) sekä työkulun aikana datalle erilaisilla muuntimilla (transformer) tehtäviin operaatioihin. FME sopii hyvin erilaisten datalähteiden lukemiseen, datan esikäsittelyyn ja kirjoittamiseen keskitettyyn tietokantaan. FME:een on mahdollista tehdä omia muuntimia yhdistelemällä olemassa olevia muuntimia tai kirjoittamalla uusia Pythonilla, JavaScriptillä, TCL:llä tai R:llä dynaamisesti tai vaihtoehtoisesti käännettävinä laajennuksina Javalla tai C++:lla (Safe Software). Tässä projektissa käytettiin Pythonia kustomoitujen muuntimien toteutuskielenä siltä osin, kuin niitä oli tarpeellista luoda.

Havaintodataa luettaessa FME-prosessi ajastettiin käynnistymään tunnin välein, ja muodostettiin kyselyä varten XML-dokumentti, jossa määriteltiin pyyntö ajoneuvojen havaintojen hakemiseksi edellisen tunnin ajalta. Pyyntö lähetettiin rajapinnalle, josta saatiin XML-muotoinen vastaus. Vastaus luettiin ja jäsennettiin siten, että jokainen XML:n pistettä kuvaava elementti muunnettiin FME:n omaksi spatiaaliseksi pistetyypiksi, jonka attribuuteiksi XML-elementtien attribuutit siirrettiin. Pisteille suoritettiin koordinaatistomuunnos Vantaan kaupungin käyttämään ETRS-GK25-koordinaattijärjestelmään, ja selvästi virheelliset havainnot poistettiin (esimerkiksi kaukana kaupungin rajojen ulkopuolella sijaitsevat). Lopuksi data kirjoitettiin PostGIS-tietokantaan attribuutti- ja geometriatietoineen. Tässä tutkimuk-

sessä käytettiin talven ja kevään 2016 aikana kerättyä dataa. Yhteensä havaintopisteitä tarkasteltavalta ajalta oli käytettävissä noin 2 200 000 kappaletta 37 eri ajoneuvolta. Näistä kuorma-autoja oli 15, henkilöautoja 13 ja erilaisia pyöräkuormaajia tai traktoreita 9.

## 6.2 Tilatiedon automaattinen määrittäminen

Katujen kunnossapidon tilannetiedon määrittämiseksi on tärkeää tietää, mitä koneet tekevät milläkin hetkellä. Kaikkien relevanteimmat statukset, joita koneet voivat saada ovat siirtyminen, auraus, suolaus ja hiekoitus. Kesäkunnossapitoon kuuluu lisäksi esimerkiksi katujen harjaus, pesu ja pölynsidonta. Koska rajapinnasta saatavassa datassa ei kerrota suoraan millä statuksella koneet toimivat havainnon hetkellä, täytyy status määrittää prosessin aikana seuraamalla koneiden kanavatietojen muutoksia. Koneilla on neljä kanavavaihtoehtoa ja ne saavat aktivoituessaan arvon 1 ja pois päältä mennessään arvon 0. Mikäli kanavalle ei tapahdu muutosta, ei mitään niihin liittyvää tietoa välity. Oikean tilatiedon päättämiseksi on siis tiedettävä, mitä kunkin koneen kanavat ovat ja sen lisäksi seurattava niiden arvojen muutoksia.

Konekohtaiset kanavakartat voidaan hakea toisella rajapintapyynnöllä ja ne tallennetaan samaan tietokantaan kuin havainnotkin. Jotta mahdolliset muutokset kanavatiedoissa saadaan päivitettyä ajantasaisesti, suoritetaan kanavakarttojen päivitys aina ennen datan hakemista. Kun tehdään uusi kysely rajapintaan, haetaan kullekin koneelle sen edellinen tunnettu status tietokantaan edellisillä hauilla tallennetuista havainnoista. Mikäli koneelta ei ole lainkaan havaintoja viimeisen tunnin ajalta, merkitään statukseksi tyhjä (mikä vastaa joko paikallaan oloa tai siirtoajoa). Tämän jälkeen havainnot kulkevat aikajärjestyksessä Python-aliohjelman läpi, joka merkitsee tilatietojen muutokset tietorakenteeseen ja päivittää sitä sekä merkitsee jokaiselle havainnolle niiden epookin mukaisen viimeisimmän tilatiedon. On huomattavaa, että koneilla voi olla useampia aktiivisia tilatietoja yhtäaikaaisesti ja kaikkien muutoksia on seurattava yhden ajon aikana. On esimerkiksi täysin mahdollista, että ajoneuvo auraa ja hiekoittaa yhtäaikaaisesti (Tiittanen 2016).

Kun statukset on saatu selvitettyä koko havaintopistejoukolle, kirjoitetaan tiedot tietokantaan. Havainnot kirjoitetaan omaan tauluunsa ja havainnon status omaansa, ja näiden välille muodostetaan yhteys käyttämällä statustaulun viiteavaimena havainnot-taulun pääavainta. Tämä jako tehdään, jotta mahdollistetaan yhdestä-moneen (1-*m*) relaatiot havaintojen ja tilatietojen välille, jotta koneille voidaan tallentaa useita yhtäaikaaisia statuksia. Liitteessä 2 on esitetty graafinen kuvaus järjestelmässä käytetyistä tietokantatauluista.

Kanavatiedot määrittyvät Vantaan telemetriajärjestelmässä automaattisesti siten, että esimerkiksi hydraulisesti toimivaan aisaan on kiinnitetty rajakytkin, joka aktivoituu, kun aisa lasketaan tarpeeksi alas. Järjestelmään on merkitty erikseen kanavakartat jokaiselle koneelle, jotka kertovat mitä kyseisen kanavan aktivoituminen tarkoittaa. Esimerkiksi talvella etunos-timen rajakytkimen aktivoituminen pyöräkuormaajalla tarkoittaa auraamista. Vastaavasti koneen taakse kytkettävän lisälaitteen aktivoituminen tarkoittaa esimerkiksi hiekoitusta. Tämän kaltaisessa järjestelmässä syntyy virheitä tilatietoihin, mikäli koneella käytetään jotain muuta lisälaitetta, kuin mikä kanavakarttaan on määritetty. Sama kone käyttää lumisena aikana etuaisean kiinnitettyä auraa, ja lumettomana aikana esimerkiksi harjauslaitetta. Kanavatiedot voi päivittää automaattisesti esimerkiksi päivämäärän mukaan jakamalla vuosi kesä- ja talvikausiin, tai järjestelmän ylläpitäjä voi huolehtia siitä manuaalisesti. Kummatkin

menetelmät ovat kuitenkin alttiita virheille: päivämäärään perustuvassa järjestelmässä kesä- ja talvikausien epäselvä raja aiheuttaa syys- ja kevätaikaan väärin tilatietojen määrittämiä, ja toisaalta jonkun on päätettävä missä kohtaa kaudet vaihtuvat. Manuaalinen päivitys taas on virhealtista siksi, että tiedonvälitykseen voi syntyä herkästi katkoksia, ja käsin tietojärjestelmiin tehtävät päivitykset ovat usein virhealttiita.

Statuksen määrittämistä tarkemmin lisälaittekohtaisesti on pohdittu Vantaalla. Kaikissa elektronisissa tai mekaanisissa toteutustavoissa on haasteena olosuhteet, joissa laitteita joudutaan käyttämään. Kova pakkanen ja toisaalta märät lumikelit rajaavat pois halvimmat kulluttajatuotteet, sillä komponentit ja liittimet tarvitsevat vähintäänkin kestävä ja tiiviin koteloinnin selvitäkseen kosteudesta ja jäätymisestä. Käyttäjille ei myöskään voi seurata kohtuutonta lisävaivaa järjestelmän käyttämisestä. Käytännössä laitteita on voitava käyttää pimeässä, paksut rukkas kädessä, joten mikäli esimerkiksi lisälaitteiden liittimiä täytyy kytkeä käsin, on niiden oltava riittävän suuria. Toisaalta vaatimus ei ole myöskään kohtuuton, sillä lisälaitteet kytketään jo nyt koneen hydrauliseen ja sähköiseen järjestelmään useilla eri liittimillä. Komponenttien on myös kestävä suurta mekaanista kulutusta, jota niihin väistämättä kohdistuu lisälaitteita vaihdettaessa ja esimerkiksi jäätä poistettaessa.

Yksi tilatietojen tarkempaan määrittämiseen ja käytettävien lisälaitteiden tunnistukseen sopiva menetelmä on esimerkiksi ylimääräinen sähköjohdolla kytkettävä silmukka lisälaitteen läpi. Pistokevalinta taas koneen päässä määrittelee lisälaitteen tyypin. Toinen vaihtoehto on RFID-lukija (Radio Frequency Identification) koneessa ja RFID-tunniste lisälaitteella. RFID on radioaaltoihin perustuva etätunnistusmenetelmä, joka koostuu lukijasta ja tunnistimesta. Lukija skannaa radioaaltoja käyttäen tunnistetta, joka saa indusoitumisesta tarvitsemansa sähkövirran ja lähettää omalla antennillaan tunnistenumeron. RFID-vaihtoehdon etuna on helppous, sillä lisälaitte kiinnitetään koneeseen normaalisti, ja koneessa kiinni oleva lukija lukee tunnisteen etänä ilman, että käyttäjän tarvitsee huolehtia siitä. Tunnistimet ovat halpoja, mutta lukijat maksavat enemmän. Metallia vasten kiinnitettyä tavallista RFID-tunnistetta ei kuitenkaan voida lukea luotettavasti, mikä saattaa tuottaa ongelmia sopivan kiinnityskohdan löytämisessä kaikissa laitteissa. Kalliimmat tunnistimet voivat kuitenkin toimia metallipinnoille kiinnitettynäkin. Huonona puolena voidaan pitää myös epävarmuutta tunnistimien ja lukijoiden sääolojen ja mekaanisen rasituksen kestävydestä. Helsingin kaupungin katujen kunnossapidosta suurelta osin vastaava Stara on kokeillut osassa kalustostaan RFID-tekniikkaan perustuvaa lisälaitteiden tunnistusta. Sen huonoina puolina pidettiin komponenttien hintaa sekä kustannuksia, joita komponenttien hajoaminen aiheuttaa etenkin välillisesti, kun koneita joudutaan viemään varikolle korjattavaksi pelkän telemetriaan liittyvän sensorin hajoamisen johdosta (Laukkanen 2015).

Kolmas vaihtoehto on kuljettajan manuaalisesti käytettävät tavat ilmoittaa kulloisestakin tilatiedosta. Näitä ovat paino- tai kääntökytkimet hytissä tai erillisellä päätelaitteella (esimerkiksi älypuhelimella) käytettävä sovellus. Näiden menetelmien selkeänä huonona puolena on se, että ajoneuvon kuljettajan on muistettava käyttää niitä oikein. Stara käyttää Geometrix Oy:n mobiilisovellusta muun muassa tilatietojen merkitsemiseen. Sovellus toimii samalla tuotannonohjausjärjestelmänä ja yleisenä apuvälineenä, joten käyttäjälle on olemassa selkeä motiivi sen jatkuvalle käytölle (Laukkanen 2015). Käyttäjille pitäisikin olla tällaiseen menetelmään turvauduttaessa olla jotakin konkreettista hyötyä sovelluksen käyttämisestä, jotta syy järjestelmän käyttöön ei johdu vain työnjohdollisesta ohjeesta ja näyttäytyä näin epämotiivisena ylimääräisenä rasitteena.



### 6.3 Järjestelmän havaintopisteet

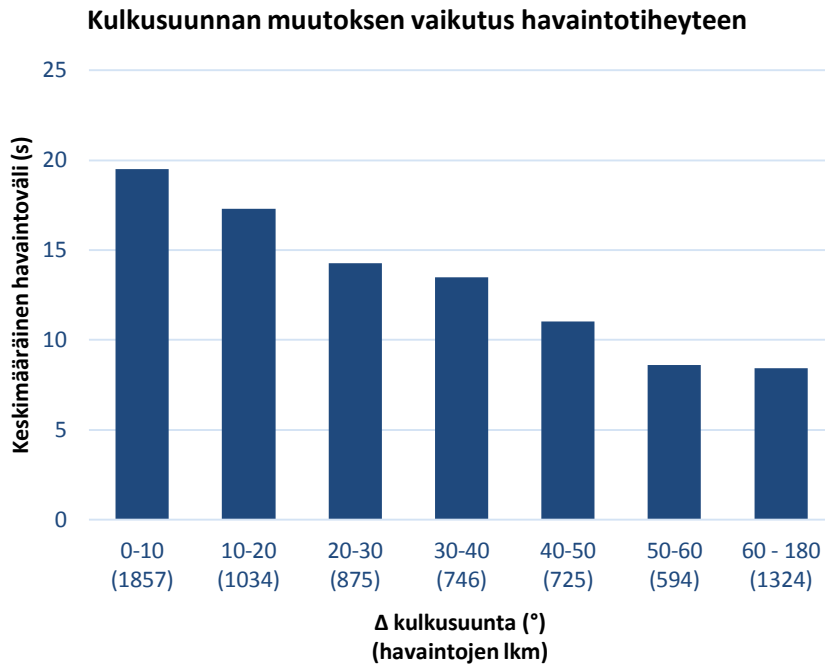
Järjestelmän keräämien paikkatietojen sijaintitarkkuutta testattiin kenttäkokeella. Telemetriailaitteistolla varustettuun pakettiautoon kytkettiin lisäksi RTK-vastaanotin, joka käytti mittaamiseen Trimblen VRS-verkkoa. Vastaanotin asetettiin mittaamaan sijaintiaan 1 Hz frekvenssillä eli havaintoväli oli 1 sekunti. Tällaisella RTK-mittauksella saavutetaan parempi kuin 10 cm sijaintitarkkuus, joten mittausdataa voidaan pitää tarkkana referenssiaineistona. Telemetriarajapinnasta poimittiin kyseisen pakettiauton tuottamat liikeradat, ja niitä verrattiin RTK-mitattuihin liikeratoihin. Vertailut toteutettiin kahtena eri päivänä. Yhteensä ajojen pituudeksi tuli 55,8 km ja ne koostuivat ajoneuvon paikannuksella 318 havaintopisteestä ja RTK-mittauksella 4918 havaintopisteestä. Tuloksena saatiin telemetriajärjestelmän sijaintitarkkuus ja keskimääräinen havaintoväli. Vertailun tulokset on esitetty taulukossa 3.

|                                       | <i>Vertailuajo<br/>4.4.2016</i> | <i>Vertailuajo<br/>14.4.2016</i> | <i>Yhteensä</i> |
|---------------------------------------|---------------------------------|----------------------------------|-----------------|
| Liikeradan pituus                     | 17,5 km                         | 38,3 km                          | 55,8 km         |
| Havainnot (RTK)                       | 2158                            | 2760                             | 4918            |
| Havainnot (Ajoneuvo)                  | 105                             | 213                              | 318             |
| Keskivirhe                            | 5,02 m                          | 5,60 m                           | 5,41 m          |
| Mediaanivirhe                         | 3,90 m                          | 4,62 m                           | 4,09 m          |
| Keskihajonta $\sigma$                 | 4,54 m                          | 4,93 m                           | 4,81 m          |
| $2\sigma$ (95 %)                      | 9,08 m                          | 9,86 m                           | 9,62 m          |
| Ajoneuvon keskimääräinen havaintoväli | 20,55 s                         | 13,00 s                          | 15,4 s          |

*Taulukko 3: Vantaan telemetriajärjestelmän paikannusdatan vertailu referenssiaineistoon*

Tulokset sopivat sijaintitarkkuuden osalta melko hyvin yhteen kirjallisuuden kanssa. Newson ja Krumm (2009) arvioi GPS-paikannuksen virheen keskihajonnaksi 4,5 m, ja monet muut ovat käyttäneet samansuuntaisia arvoja. GPS-järjestelmän julkisen paikannuspalvelun (SPS) laatustandardin mukaan sijaintitarkkuuden (95 %) pitäisi olla alle 13 m, mikä oli myös vertailun tulosten kanssa linjassa oleva arvo (Yhdysvaltain puolustusministeriö 2008).

Havaintovälin osalta dataa voidaan pitää kirjallisuuteen nähden keskitiheänä. Kirjallisuudessa harvoina liikeratoina pidetään yleensä 30 s ja sitä suurempien havaintovälien dataa. Toisaalta tiheinä liikeratoina pidetään yleensä 1–5 s havaintovälin dataa. Havaintoväliä tarkastellessa on huomioitava, että Vantaan ajoneuvopaikannusjärjestelmän pitäisi lähettää ja tallentaa dataa adaptiivisesti riippuen siitä, onko ajoneuvo kääntymässä vai liikkumassa suoralla linjalla. Tämän todentamiseksi liikeratoja analysoitiin poimimalla tietokannasta selvästi liikkeessä olevien ajoneuvojen dataa. Satunnaisotannalla muodostettiin 120 liikerataa konekohtaisista havaintopisteistä, joiden aikaero oli korkeintaan 60 s ja jotka muodostivat vähintään 750 metriä ja 20 pistettä pitkiä liikeratoja. Ajoneuvot jaettiin ryhmiin sen perusteella, kuinka monta astetta niiden kulkusuunta oli muuttunut kahden edellisen havaintopisteen aikana. Tutkittavat liikeradat koostuivat yhteensä 7156 havaintopisteestä. Analyysin tulokset on esitetty kuvassa 11. Tulosten perusteella havaintoväli vaihtelee välillä 8–20 s riippuen ajoneuvon suunnanmuutoksen suuruudesta.



Kuva 11: Vantaan telemetriajärjestelmän havaintofrekvenssin muutokset suhteessa hetkelliseen kulkusuunnan muutokseen.

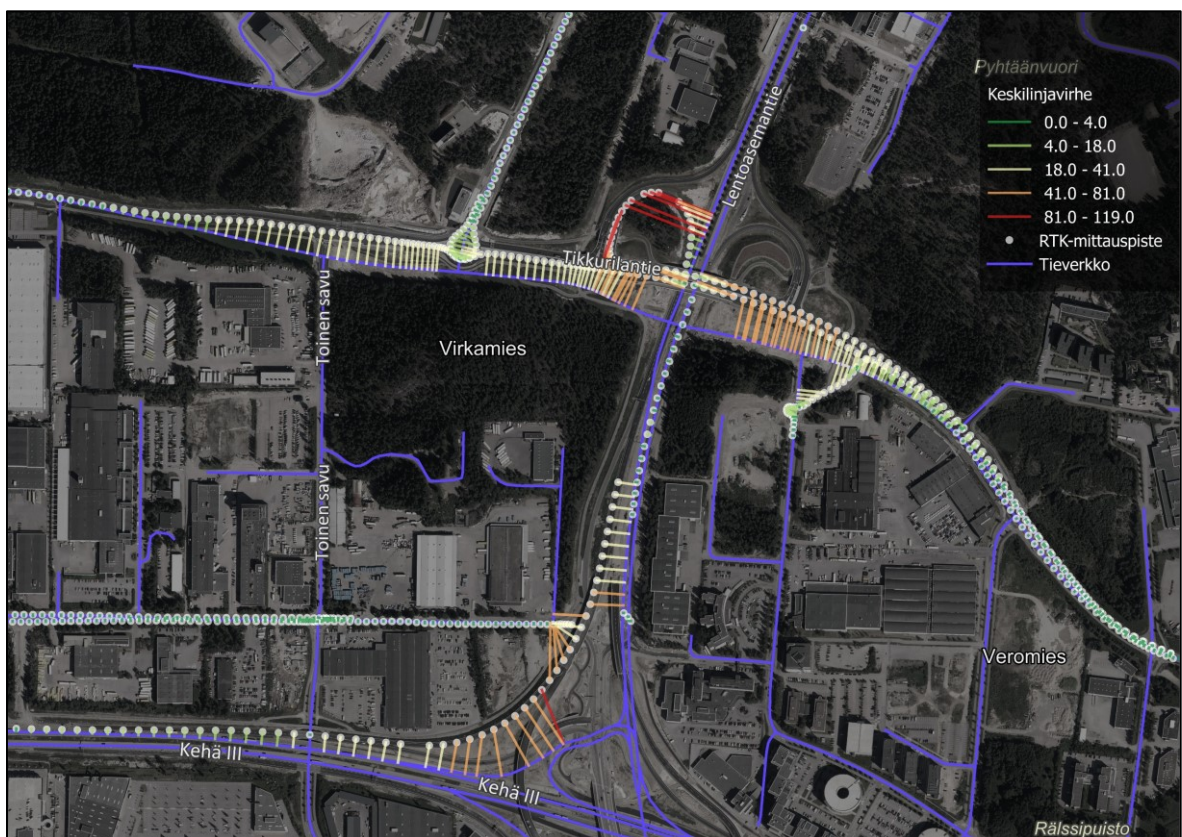
Datan havaintovälin analyysi tuki näkemystä, jonka mukaan tämän tutkimuksen karttasovitusongelman ratkaisuun ei voida soveltaa tiheää dataa vaativia topologisia menetelmiä, vaan reittien esilaskentaan perustuvia harvalle datalle sopivia menetelmiä.

## 6.4 Tieverkko

Digitaalisena karttana tutkimuksessa käytetään Suomen kansallista tie- ja katuverkon tietojärjestelmää Digiroadia, jonka ylläpidosta vastaa Liikennevirasto. Tietosisältönä tieverkon kohteilla on lukuisia tietä kuvaavia ominaisuustietoja. Digiroadin laatumallissa laatutavoite karttasovitusongelman kannalta relevanteille ominaisuustiedoille on virheellisten tietojen suhteellisella määrällä mitattuna 0,01 % tien nimelle ja liikennevirran suunnalle, 1 % nopeusrajoitukselle, 0,002 % liikenne-elementtien kattavuudelle (puute/ylimääräisyys), 2 % kaistojen lukumäärälle ja 2 % tien leveydelle. Tien leveys- ja nopeusrajoitustietojen luvataan olevan kattavia vain maanteiden osalta (Digiroad laatumalli 2012). Aineisto on topologisesti ehyttä ja pääsääntöisesti muodostettu Maanmittauslaitoksen maastotietokannan pohjalta. Maastotietokannan nimellismittakaava on 1:5000–1:10000. Käytännössä tiekohteiden sijaintitarkkuus on 3 metriä 95 % luottamusvälillä (Maastotietojen laatumalli).

Lähitulevaisuuden toisena vaihtoehtona on Vantaan kaupungin oma keskilinja-aineisto, jonka mallintaminen ja ylläpito on aloitettu vuonna 2016 ja joka valmistunee vuoden 2016 loppuun mennessä. Etuina nykyiseen Digiroad-aineistoon on keskilinjien parempi sijaintitarkkuus, sillä digitointi suoritetaan hyvin tarkoilta ilmakuvilta ja 1:500 nimellismittakaavaisen kantakartan pohjalta, parempi kattavuus erityisesti pienten reittien ja kevyen liikenteen väylien osalta sekä ennen kaikkea aineiston parempi ajantasaisuus. Toisaalta projektissa syntyvä data siirtyy todennäköisesti keskipitkällä aikavälillä osaksi Digiroadia.

Tieverkon keskilinjamallinnuksesta johtuvan keskilinjavirheen suuruuden määrittelemiseksi RTK-mitattuja liikeratoja päätettiin verrata myös tieverkkoon. Jokaiselle RTK-mitatulle pisteelle haettiin lähin segmentti tieverkosta ja selvitettiin etäisyys tähän segmenttiin. Vertailua tehtäessä havaittiin, että liikennejärjestelyt olivat jo ehtineet muuttua osalla RTK-mitatuista reiteistä: Lentoasemantien risteys- ja ramppijärjestelyt sekä Tikkurilantien että Kehä III:n välillä olivat muuttuneet, samoin kuin koko Tikkurilantien linjaus lentokentän eteläpuolella. Muutokset puuttuivat vielä käytettävästä digitaalisesta tieverkosta, joten dataa jouduttiin poistamaan näiltä osin, jotta keskilinjavirhe pystyttiin selvittämään. Toisaalta esimerkki osoitti, miten rakentuvilla alueilla digitaaliset paikkatietokannat vanhentuvat nopeasti, mistä seuraa välttämättä karttasovituksen tarkkuuden heikentymistä. Mahdollisimman ajantasaisen tieverkon saaminen käyttöön on tärkeää kaikissa karttasovitusta hyödyntävissä sovelluksissa. Tieverkon muutokset ja niiden vaikutukset keskilinjavirheeseen on esitetty kuvassa 12. Kuvassa taustalla olevassa ilmakuvassa tielinjaukset ja ramppirakenteet vastaavat todellisuutta, mutta digitaalista tieverkkoa ei ole päivitetty vastaamaan sitä.



Kuva 12: Digitaalisen kartan ajantasaisuuden puutteiden aiheuttamat erot keskilinjavirheeseen.

Karkeat epäajantasaisuudesta johtuvat virheet poistettiin (436 pistettä), jonka jälkeen jäljelle jäi 4482 pistettä. Keskimääräiseksi keskilinjavirheeksi saatiin 2,38 m, kun se virheellisellä datalla oli lähes 4 m. Visuaalisen tarkastelun perusteella keskilinjavirheet olivat suurimmillaan monikaistaisilla teillä ja erityisesti suurilla risteysalueilla ajettaessa. Tämä onkin järkeenkäypää, kun huomioidaan, että risteykset mallinnetaan risteyskeskelle yhdistyvillä keskilinjajoilla, joita pitkin risteyksissä liikkuvat ajoneuvot eivät kuitenkaan käytännössä kulje valitessaan loivemman reitin risteyskeskelle. Tämä intuitio vahvistui, kun keskilinjavirheen suuruus ryhmiteltiin tieluokittain. Tulokset on esitetty taulukossa 4. Virheen jakautumisesta

havaitaan, että se noudattelee teiden toiminnallista luokittelua kasvaen tien luokan ja siten leveyden mukaisesti.

| <i>Tieluokka</i> | <i>Keskilinjavirhe (ka.)</i> |
|------------------|------------------------------|
| Kantatie         | 2,99 m                       |
| Seututie         | 2,38 m                       |
| Kokoojakatu      | 2,25 m                       |
| Liityntäkatu     | 2,15 m                       |
| Yksityistie      | 1,40 m                       |

*Taulukko 4: Keskilinjavirheen keskiarvo tieluokittain.*

Digiroad-dataa jaetaan ESRI-Shape formaatissa sekä K- että R-julkaisusarjana. R-sarjassa hyödynnetään niin sanottua lineaarista referointia, mikä tarkoittaa ominaisuustietojen dynaamista muuttumista tieverkon linkeillä niiden pituuden suhteen. Tällaisen datan tarkasteluun ja muokkaamiseen tarvitaan tavallisesti lineaarista referointia tukeva ohjelma. K-sarjassa tielinkit on pilkottu ominaisuustiedoiltaan yhtenäisiin segmentteihin. Sovellusta varten haettiin K-julkaisusarjan data, josta säilytettiin karttasovituksen kannalta olennaiset tiedot. Data muunnettiin FME:n omaan FFS-binääriformaattiin (FME Feature Store) ja sille luotiin spatiaalinen indeksi datan alueittaisen nopean lukemisen mahdollistamiseksi.

## 7 Karttasovitusalgoritmien toteutus

Tutkimuksessa päätettiin toteuttaa kaksi vaihtoehtoista globaalia algoritmia karttasovitusongelman ratkaisemiseksi ja vertailla niiden sovitustarkkuutta ja suorituskyykyä. Ensimmäinen toteutettu algoritmi perustui Markovin piilomalliin (HMM) ja toinen toteutettu algoritmi perustui sumeaan logiikkaan. Seuraavissa luvuissa esitellään algoritmien implementointi ja niihin tehdyt viritykset. Luvussa 7.3 tehdään algoritmien vertailu referenssiaineiston avulla.

### 7.1 HMM-karttasovitus

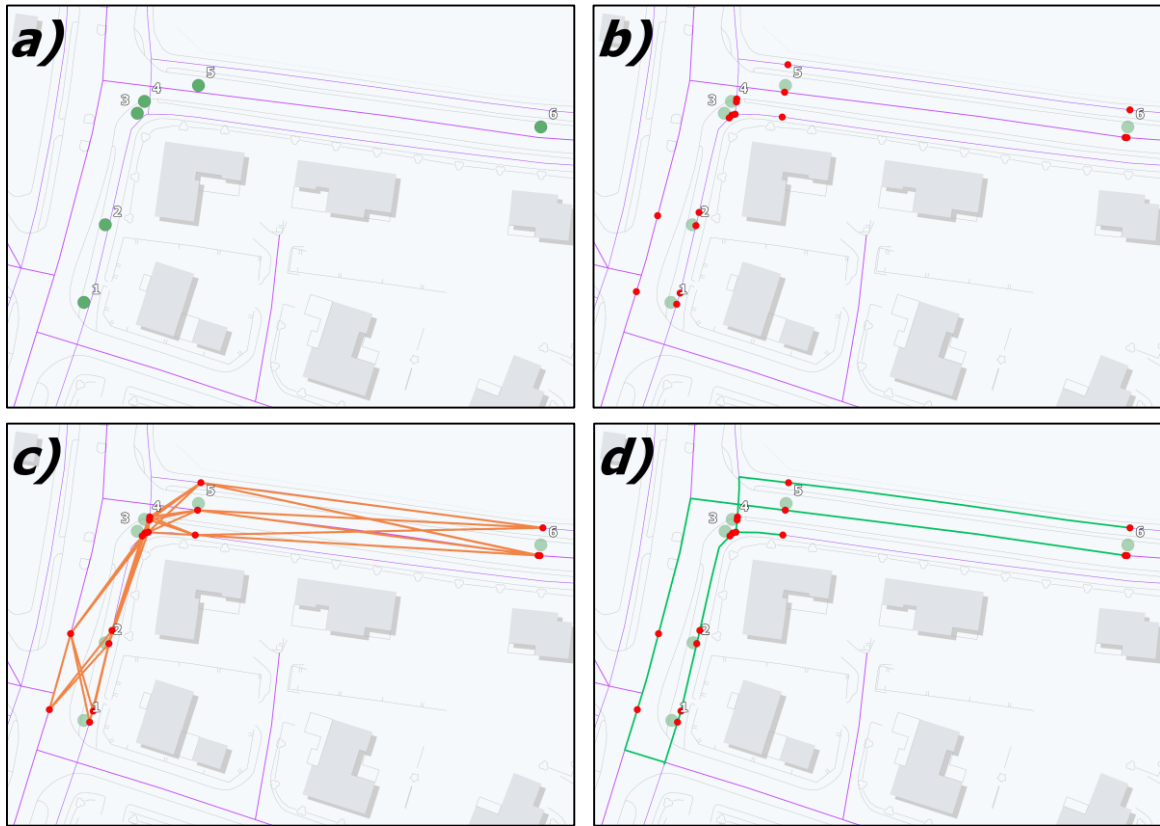
Karttasovitusalgoritmi toteutettiin FME:een kirjoittamalla kustomoitu muunnin Python-ohjelmointikielellä. Python on noussut suosituksi avoimen lähdekoodin vaihtoehdoksi tieteiliseen laskentaan etenkin SciPy-ekosysteemin (Scientific Python) ansiosta. Koska FME-kohteita voi manipuloida haluamallaan tavalla Python-muuntimella ajon aikana ja mitä tahansa kirjastoja voidaan ottaa käyttöön, oli tämä luonnollinen teknologiavalinta. Python-kirjastoja, jota tässä algoritmissa hyödynnettiin, olivat NumPy-matriisilaskentakirjasto, Matplotlib visualisointikirjasto ja Pomegranate-kirjasto tilastolliseen mallintamiseen, jossa on toteutettuna muun muassa Viterbi-algoritmi.

Karttasovitusongelma voidaan muotoilla HMM:iin siten, että erilaiset vaihtoehtoiset tiesegmentit ovat mallin tiloja. Yksi satelliittipaikannuksella mitattu piste  $p_i$  muodostaa  $k$  pistekandidaattia, jotka ovat pisteen  $p_i$  projektioita lähimmille  $k$  tiesegmentille. Nämä segmentit haetaan lähimmän  $k$ -naapurin haulla tieverkon tietomallista. Täysin epärealistiset vaihtoehdot voidaan sulkea pois rajaamalla lähimmän naapurin hakua siten, että segmentin maksimietäisyys pisteeseen  $p_i$  on enintään  $d$  metriä. Teoriassa parhaan sovitustarkkuuden takaimiseksi voitaisiin reitit hakea koko käytettävissä olevasta tieverkosta, mutta laskentaan kuuluvan ajan rajoittamiseksi parametreille on löydettävä mahdollisimman pienet arvot, joilla saavutetaan riittävä sovitustarkkuus. Quddus et al. (2006) esittelee virhe-ellipsin määrittämiseen perustuvan älykkään menetelmän lähimpien segmenttien valintaan (luku 4.1), mutta tässä tutkimuksessa käytettiin staattista arvoa johtuen ensinnäkin siitä, ettei käytettävästä tietolähteestä ollut saatavissa virhe-ellipsin laskentaan tarvittavia tietoja, ja toiseksi siitä, että tarkasta rajauksesta saatava hyöty ei ole harvalla liikeradalla haittoihin eli potentiaalisesti hausta pois jääviin oikeisiin segmentteihin verrattuna tarpeeksi suuri.

#### 7.1.1 Prosessin kulku

Prosessin esikäsittelyvaiheessa tietokannasta haettiin halutulla rajauksella (esimerkiksi aikarajaus, ajoneuvorajaus tai spatiaalinen rajaus) *havaintopisteitä*. Tieverkkoa poimittiin analyysiin mukaan havaintopisteiden alueelta muodostamalla pistejoukon konveksipeite (engl. convex hull) ja bufferoimalla sitä lisäksi 500 metriä, jotta valideja reittivaihtoehtoja pisteiden välillä ei jäänyt pois analyysistä. Pisteille haettiin tieverkolle projisoidut *pistekandidaatit* siten, että lähimmän naapurin haun parametreiksi valittiin maksimietäisyys pisteestä 70 metriä ja naapureiden lukumäärä  $k = 6$ . Pistekandidaateista muodostettiin *lähde-kohde*-parit siten, että kaikki pisteiden  $p_i, p_{i+1}$  pistekandidaattien yhdistelmät kuuluvat niihin. Näille pareille  $c_{i,k} \rightarrow c_{i+1,k}$  haettiin lyhyimmät *reittikandidaatit* tieverkosta. Kuva 13 selventää mitä

käytettävillä termeillä tarkoitetaan. Kuvassa on selkeyden säilyttämiseksi haettu vain kolme lähintä segmenttiä havaintopistettä kohden.



Kuva 13: a) Havaintopisteet (6 kpl) ja taustalla tieverkko (violetilla), b) tieverkolle havaintopisteistä projisoitunut pistekandidaatit (3 per piste, punaisella), c) pistekandidaattien väliset lähde-kohde-parit (oranssilla), d) lähde-kohde-pareista muodostetut reittikandidaatit (vihreällä).

Reittikandidaattien perusteella muodostettiin transitiomatriisi, joka kuvaa siirtymätodennäköisyyksiä kaikkien pistekandidaattien välillä. Tämä matriisi on erittäin harva, eli suurin osa matriisin alkioiden arvoista on nolla, johtuen siitä, että tilasiirtymät ovat mahdollisia ainoastaan kahden peräkkäisen havaintopisteen pistekandidaattien välillä. Emissiomatriisin arvot määriteltiin todennäköisyysjakaumasta, joka kuvaa paikannuksen virhettä. Havaittu sekvenssi puolestaan määriteltiin havaintopisteen etäisyytenä segmenttikandidaatteihin. Mallin aloitustodennäköisyydet määriteltiin jakamalla kaikille pisteen 1 kandidaateille todennäköisyys  $1/k$ . Emissio- ja transitiomatriisi sekä aloitustodennäköisyyksien matriisi muodostaa HMM:n, josta voidaan etsiä tietyn havaitun sekvenssin posterioritodennäköisyydet maksimoiva tilojen ketju käyttämällä Viterbi-algoritmia. Lopuksi tätä tilaketjua vastaavat reitit poimittiin kaikkien reittikandidaattien joukosta ja valittiin jatkoon.

### 7.1.2 Emissio- ja transiotodennäköisyyksien määrittäminen

Tieverkolle projisoitujen pistekandidaattien  $c_{i,k}$  euklidiset etäisyydet pisteestä  $p_i$  saavat emissiotodennäköisyyden, joka perustuu satelliittipaikannuksen sijaintitarkkuuteen. Newsonia ja Krummia (2009) mukaillen todennäköisyys määriteltiin nollakeskiarvoisesta normaalijakaumasta. Newson ja Krumm (2009) käyttivät keskihajontana paikannuksen keskivirheeksi

arvioitua 4,5 metriä. Tässä tutkimuksessa käytettiin laskettua paikannusvirheen keskihajontaa  $\sigma_1 = 4,8$  m ja keskilinjavirhettä  $\sigma_2 = 2,3$  m. Jakauman keskihajonnaksi tuli täten  $\sigma = \sigma_1 + \sigma_2 = 7,1$  m. Emissiotodennäköisyydet määriteltiin siis nollakeskiarvoisesta normaalijakaumasta

$$p = \frac{1}{\sqrt{(2\pi)\sigma_z}} e^{-0,5 \left(\frac{d}{\sigma_z}\right)^2} \quad (5)$$

missä keskihajonta  $\sigma = 7,1$  ja

$d$  on euklidinen etäisyys havainnon  $p_i$  ja pistekandidaatin  $c_{i,k}$  välillä.

Kuten edellisessä luvussa mainittiin, jokaiselle pistekandidaatille  $c_{i,k}$  muodostettiin pari kaikkien seuraavan pisteen  $p_{i+1}$  pistekandidaattien  $c_{i+1,k}$  kanssa. Yhteensä pistekandidaateista muodostuu siis  $kP$  ja kandidaattipareja  $k^2(P-1)$  kappaletta. Kandidaattiparien pistetiedoista voidaan laskea niiden välinen euklidinen etäisyys ja aikaero. Transitiotodennäköisyys määriteltiin euklidisen etäisyyden  $d$  ja tieverkkoa pitkin kulkevan lyhyimmän reitin pituuden  $d_{path}$  erona. Absoluuttisen eron sijaan käytettiin kuitenkin suhteellista eroa  $\Delta_{path} = d / d_{path}$  kuten Lou et al. (2009) ja Koller et al. (2015). Tämä suhteellinen pituusero on oikeille reiteille yleensä lähellä yhtä, eli transitiotodennäköisyys voidaan mallintaa eksponentiaalisena jakaumana

$$p = \frac{1}{\beta} e^{-\Delta_{path}/\beta} \quad (6)$$

missä  $\Delta_{path}$  on edellä mainittu suhteellinen pituusero ja  $\beta$  on jakauman muotoa kontrolloiva parametri.

Pienet  $\beta$ -arvot aiheuttavat transitiotodennäköisyyksien putoamisen hyvin lähelle nollaa sellaisilla kandidaattipareilla, joilla on pienet  $\Delta_{path}$ -arvot. Tästä seuraa, että mallissa transitiotodennäköisyydet saavat suuremman painoarvon verrattuna emissiotodennäköisyyksiin, ja toisaalta myös sen, että malliin tulee helpommin katkoksia, joissa kaikki pisteen transitiotodennäköisyydet ovat lähes nolla. Suuret  $\beta$ -arvot vastaavasti kasvattavat emissiotodennäköisyyden painoarvoa, sillä pienemmät  $\Delta_{path}$ -arvot tulevat hyväksyttävimmiksi. (Koller et al. 2015.)

Liikeradan havaintoväli vaikuttaa siihen, mikä parametrin  $\beta$  arvoksi kannattaa valita, sillä suurempi havaintoväli tekee reiteistä yleensä ottaen mutkaisempia. Song et al. (2012) laskivat artikkelissaan optimaalisia  $\beta$ -arvoja eri havaintoväleille ja totesivat sen olevan 10 sekunnin havaintovälille noin 4 ja 20 sekunnin havaintovälille noin 12.

EkspONENTTijakaumaa käytettäessä ja muulloinkin todennäköisyyslaskentaan liittyvien algoritmien kanssa toimittaessa on huomioitava, että hyvin pienet luvut aiheuttavat monissa ohjelmointikielissä liukulukujen alivuotoa, mikä voi aiheuttaa ongelmia erityisesti laskuissa, joissa pieniä todennäköisyyksiä kerrotaan keskenään. Alivuoto tarkoittaa sitä, että desimaaliluku saa pienemmän arvon kuin mitä ohjelma pystyy käsittelemään. Alivuoto aiheuttaa yleensä joko poikkeuksen ohjelman suorituksessa tai muuttujan arvon asettamisen nolllaksi. Jos tällaista muuttujaa käytetään myöhemmin esimerkiksi jakajana toisessa laskutoimituksessa, seuraa nolllalla jakamisesta poikkeus. (Wen 2007.)



Alivuotojen ehkäisemiseksi todennäköisyyksien kanssa hyödynnetään usein todennäköisyyden  $p$  logaritmia  $\log(p)$ . Tällöin logaritmin ominaisuuksia (esimerkiksi  $\log(p_1 p_2) = \log(p_1) + \log(p_2)$ ) voidaan hyödyntää koko algoritmin suorituksen ajan eikä alivuotoja voi tapahtua. Kun lopputulokseksi halutaan jokin ei-logaritminen todennäköisyys, voidaan  $p$  palauttaa laskemalla logaritmin eksponentti  $\exp(\log(p))$ . Toinen vaihtoehto, jolla voidaan estää sellaisten muuttujien nollaantuminen, joilla oletetaan olevan esimerkiksi positiivinen arvo, on käyttää toleranssia  $\varepsilon$ . Kaikkien muuttujien suuruus voidaan testata toleranssia vasten ja sitä pienemmät muuttujat saavat jonkin vakioarvon. (Wen 2007.)

Transitiomatriisin rivit normalisoidaan ennen Viterbi-algoritmin suorittamista siten, että ne summautuvat yhteen. Mikäli kaikki todennäköisyydet ovat kuitenkin äärimmäisen pieniä, ei normalisointia ole mielekästä tehdä niiden osalta, vaan merkitä tällaisen todennäköisyyden arvoksi 0. Tämän seurauksena transitiomatriisiin voi syntyä rivejä, jotka summautuvat nolnaan, mikä tarkoittaa sitä, että transiio kahden havaintopisteen välillä on mahdoton, eikä Viterbi-algoritmi voi tällöin kulkea HMM:n läpi.

Tällaisissa tilanteissa suoritettiin mallin katkaisu siten, että ensimmäinen malli käsittää pisteet  $p_1 \dots p_x$  jossa  $x$  tarkoittaa transitiomatriisin nolnaan summautuvaa riviä. Emissiomatriisi, transitiomatriisi ja havaintosekvenssi katkaistaan tästä kohdasta ja Viterbi-algoritmi suoritetaan parhaan tilasekvenssin löytämiseksi ja sitä vastaavat reittikandidaatit tulostetaan. Algoritmin suoritusta jatketaan tämän jälkeen siten, että aloitustodennäköisyydet  $s_0$  alustetaan osoittamaan pisteen  $x+1$  kandidaatteihin ja transiio- ja emissiomatriisien rivit  $1 \dots x$  poistetaan. Transitiomatriisin tarkistusta jatketaan, kunnes löydetään seuraava mallin katkokohta ja sama katkaisutoimenpide toistetaan. Kun tätä menetelmää toistamalla päädytään transitiomatriisin loppuun, voidaan suorittaa koko jäljellä olevalle mallille Viterbi-algoritmi ja tulostaa reitin loppuosa.

### 7.1.3 Algoritmin testaus

Jotta algoritmeja pystyttiin testaamaan niiden kehittämisen aikana, tietokannasta poimittiin satunnaisotannalla 20 liikerataa samoilla ehdoilla kuin luvussa 6.3. Puolet liikeradoista oli autojen (kuorma-auto tai henkilöauto) ja puolet työkoneiden (pyöräkuormaaja tai traktori) liikkeiden tuottamia. Nämä liikeradat sovitettiin käsin tarkistamalla ja digitoimalla vastaamaan ajoneuvojen todellisia kuljettuja liikeratoja. Mikäli liikeradassa oli niin suuria virheitä tai epäselvyyksiä, ettei sitä voinut sovittaa manuaalisesti, hylättiin kyseinen liikerata ja poimittiin uusi tilalle. Testiaineistossa tällaisia tapauksia oli yksi.

FME:llä tehtiin algoritmien kehittämiseksi erilliset vertailuprosessit, jotka mittaavat prosessin päätteeksi algoritmien sovitustarkkuuden vertaamalla sovitettujen reittien oikealle referenssireitille osuvan pituuden suhteessa algoritmin ohittamiin todellisiin reitteihin sekä sen tuottamiin ylimääräisiin sovitustarkkuuteihin. Tämä sovitustarkkuuden mittaamenetelmä on kuvattu tarkemmin luvussa 7.3. Algoritmin parametrien optimaalisten arvojen hakemiseksi ajettiin testisarjoja, joilla haarukoitiin lähimmän naapurin parametrien  $k$  ja hakuetaisyys, sekä HMM-parametrien  $\beta$  ja  $\sigma$  sopivia arvoja. Todettiin, että parhaat tulokset autojen tapauksessa saatiin  $k$  parametrilla 6 ja etäisyydellä 70 metriä. Algoritmi saavutti parhaita tuloksia  $\beta$ -arvolla 28 ja  $\sigma$ -arvolla 7,2. Työkoneiden testausaineistolla havaittiin, että korkeammilla  $k$ -arvoilla saavutettiin parempia sovitustarkkuuksia.  $k$ -arvoilla 8–10 saavutettiin korkeampia sovitustarkkuuksia kuin arvolla 6, mutta toisaalta myös suoritusajat nousivat korkeammiksi.

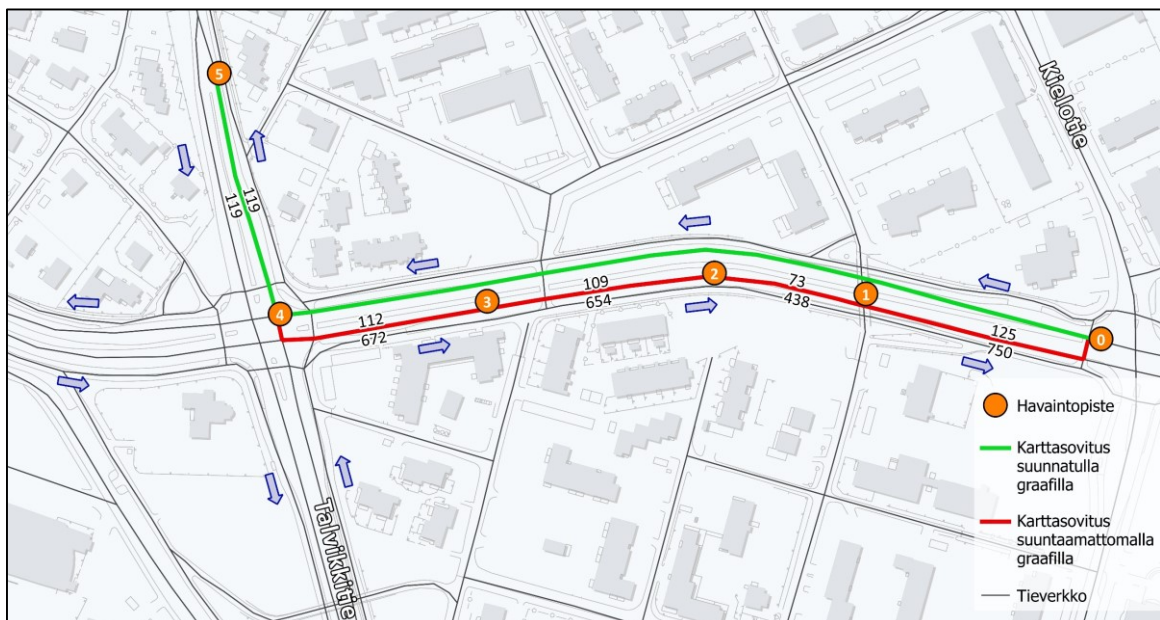


Vielä tätäkin korkeammilla  $k$ -arvoilla suoritusajat alkoivat olla kestäättömiä johtuen toisaalta transitiomatriisin kasvamisesta  $k^2P$  suuruiseksi, ja toisaalta siitä, että lyhyimpien reittien hakuja täytyy suorittaa  $k^2(P-1)$ .

#### 7.1.4 Erikoistapaukset ja viritykset

Kuten Newson ja Krumm (2009), myös tässä algoritmissa tehtiin järkevyytarkistus reittivaihtoehdon nopeusvaatimukselle ottaen huomioon havaintojen aikaleimat. Reitin nopeusvaatimus voidaan laskea jakamalla sen pituus niiden havaintopisteiden aikaleimojen erotuksella, joista se muodostuu. Mikäli siirtyminen reittiä pitkin vaati ajoneuvon keskinopeudeksi yli 40 m/s, merkittiin sen transitiotodennäköisyys nolaksi.

Yksi huomattava virhe, jonka olemassaolo tiedostettiin jo etukäteen, oli havaintojen sovitussuunnat vääristyneet ajoradalle sellaisissa tilanteissa, joissa keskikorokkeella erotetut ajoratat oli mallinnettu kahdella keskilinjaviivalla. Tällaisia ovat esimerkiksi useat Vantaan pääkadut, kuten kuvassa 14 näkyvä Valkoisenlähteenkatu. Tällaisten tapauksien korjaamiseksi oli tarpeen ottaa huomioon digitaalisen tieverkon ajosuunnat. Digiroad-aineistossa ajosuunta ilmoitetaan arvojoukolla  $\{molempiin\}$  suuntiin,  $\{digitointisuuntaan\}$ ,  $\{digitointisuuntaa\}$  vastaan}. Graafi, joka muodostetaan lyhyimpien reittien hakemiseksi, täytyi siis mallintaa suunnattuna, siten että molempiin suuntiin sallittu kaari  $n_1 \rightarrow n_2$  lisättiin graafiin myös toiseen suuntaan  $n_2 \rightarrow n_1$ . Koska katujen kunnossapidossa ei ole täyttä varmuutta siitä, että ajosuuntia välttämättä noudatettaisiin kaikissa todellisissa tilanteissa, haluttiin kaarien käyttäminen sallia edelleen toiseenkin suuntaan, mutta sellaisista siirtymistä tehtiin lähtökohtaisesti epäsuotuisia. Tämä onnistui mallintamalla jokainen graafin kaari kaksisuuntaisena, mutta asettamalla yksisuuntaisille kaarille käänteisen kulkemisen kustannus kuusinkertaisena verrattuna oikeaan kulusuuntaan.



Kuva 14: Karttasovitus a) pituuden mukaan painotetulla graafilla (punainen), sekä b) pituuden ja ajosuunnan mukaan painotetulla graafilla (vihreä). Virheellisen karttasovituksen segmenttien yhteenlasketut painot suuntaamattomana ja suunnattuna on esitetty viivan ylä- ja alapuolella vastaavasti.

Ajosuuntien mallintamisesta graafiin seurasi myös ongelmia sellaisissa tapauksissa, joissa ajoneuvo liikkuu pienellä alueella tai seisoo paikallaan esimerkiksi liikennevaloissa. Satelliittipaikannuksen epävarmuudesta aiheutuva sijaintiratkaisujen hajonta aiheutti näissä tilanteissa pisteiden siirtymistä epäloogiseen järjestykseen liikeradan yleissuunta ja aika huomiioon ottaen. Tämä pudotti yllättävissä kohdissa transiititodennäköisyyttä pienemmäksi ja sai aikaan sen, että algoritmi löysi paikoin jonkin muun kuin oikean reitin todennäköisimmäksi vaihtoehdoksi. Ongelman korjaamiseksi liikeradan lähekkäin olevan pisteet sulautettiin yhteen, mikäli ne sijaitsivat paikannusepävarmuuden 95 % luottamusvälin (noin 9,8 m) etäisyydellä toisistaan ja ovat myös ajallisesti lähekkäisiä (enintään 60 s etäisyys). Yhdistetyn pisteen ominaisuustiedot muodostettiin sulautettujen pisteiden attribuuttien keskiarvoista.

Koska ainoastaan työkoneet liikkuvat kevyen liikenteen väylillä, suunniteltiin prosessi siten, että työkoneita varten muodostetaan graafi, jossa on mukana koko tieverkko ulkoilureitit ja kevyen liikenteen väylät mukaan lukien, mutta autoja varten käytetään graafia, joka muodostetaan pelkästään katuverkosta. Kevyen liikenteen väylien mukanaolo on karttasovituksen onnistumisen kannalta erittäin haastavaa, ja niiden käyttö kannattaakin ehdottomasti rajata vain sellaisille ajoneuvoille, joiden oletetaan käyttävän niitä. Tässä tutkimuksessa tämä toteutettiin rakentamalla FME:ssä kaksi erilaista graafia tieverkosta ja luokittelemalla liikeradat sen mukaan, minkä tyyppinen ajoneuvo on kyseessä.

Ajoneuvot, jotka kulkevat tieverkon ulkopuolella aiheuttavat tämän kaltaisille HMM-algoritmeille jonkin verran ongelmia. Ne saattavat aiheuttaa sovituksen väärille reiteille tai vähintäänkin ylimääräistä laskentaa mallin korjaamiseksi, kun sitä on jaettava pienempiin osiin. Vantaan ajoneuvot pysäköidään yleensä yöksi varikkoalueiden pihoille, ja valtava osa ajetuista reiteistä joko alkaa tai päättyy jollekin kaupungin varikolle tai tukikohtaan. Nämä reittien alut ja loput aiheuttivat virheellisiä tuloksia reittien alku- ja loppupäihin varikoiden tienoille. Erityisesti tämä koski työkoneiden karttasovitusta, koska siirtyminen varikolle tieverkolta aiheutti paikoin algoritmille tendenssiä sovittaa reittien alku- ja loppuosat kevyen liikenteen väylille, joita varikoiden lähellä kulkee. Ongelman korjaamiseksi tunnistettiin kaksi lähestymistapaa: joko tieverkko pitää mallintaa kattamaan kaikki yleisimmät alueet, joilla työkoneet liikkuvat, tai havaintopisteet pitää jättää huomioimatta näiltä alueilta. Koska digitaalista tieverkkoa halutaan päivittää vähintään kerran vuodessa ajantasaisuuden varmistamiseksi, päätettiin toteuttaa jälkimmäinen vaihtoehto. Yksinkertaisin ratkaisu oli digitoida polygonit rajaamaan tunnistettuja alueita ja suodattaa havaintopisteet pois niiden sisältä ennen kuin karttasovitus tehdään.

Alueiden tunnistaminen oli suhteellisen yksinkertaista sen jälkeen, kun ajoneuvoista oli kerätty dataa riittävän kauan. Tietokannasta tehtiin 100 000 pisteen satunnaisotanta hakemalla SQL-kyselyllä ensimmäiset 100 000 riviä, kun tulokset oli järjestetty satunnaiseen järjestykseen PostgreSQL:n `random()`-funktioilla. Tulosta ei rajattu muuten ja pisteistä muodostettiin QGIS-paikkatieto-ohjelmassa tiheyskartta (heatmap), jota visuaalisesti tarkastelemalla voitiin havaita tieverkon ulkopuoliset alueet, joilla ajoneuvoja havaitaan tavanomaista enemmän. Kuvassa 15 on esitetty pisteiden tiheyskartta. Poistettavia pistetihentymiä, joita kuvasta voidaan löytää ovat ainakin lumen vastaanottopaikka Vehkalanmäellä sekä Martinlaakson tukikohta-alueet.



Kuva 15: Havaintopisteiden tiheyskartta.

Lyhyimpien reittien laskentaan perustuvat karttasovitusmenetelmät rajaavat ajoneuvojen liikkumisen aina vain tieverkolle. Tämä aiheuttaa ongelmia, kun sovelluskohteena on katu-  
jen kunnossapito. Työkoneet, jotka hoitavat sekä katuja että kevyen liikenteen väyliä ja ul-  
koilureittejä, tekevät siirtymiä etenkin tieltä kevyen liikenteen väylälle (ja toisin päin), mistä  
tahansa siihen soveltuvasta kohdasta. Digiroadin liikenneverkon topologiassa on yleensä yh-  
teys kevyen liikenteen väyliltä kaduilla suojateiden kohdalla. Vaikka suojatiekohdat ovatkin  
usein käytössä myös kunnossapitokaluston kadun ja kevyen liikenteen väylän välillä tapah-  
tuvissa siirtymissä, eivät ne kata läheskään kaikkia siirtymiä. Sellaisissa tapauksissa, joissa  
tieverkko ei salli siirtymää, minkä työkoneen kuljettaja todellisuudessa tekee, karttasovitus-  
algoritmi tekee joko epäintuitiivisen koukkauksen lähimmän mahdollisen yhteyden kautta,  
tai malli katkeaa, koska jatkaminen on liian epätodennäköistä ja se alustetaan uudelleen.  
Katkeamisvaihtoehto on lähempänä todellisuutta, mutta koukkaukset ovat ongelmallisem-  
pia. Lisäksi liikeratojen katkeamisia halutaan yleensä ottaen välttää, joten algoritmia ei ha-  
luttu virittää siten, että jokin yksittäinen erikoistapaus ratkaistaisiin liikeradan katkaisulla.

Ongelman ratkaisemiseksi algoritmiin lisättiin mahdollisuus vapaan tilan siirtymien tekemi-  
seen. Vapaan tilan siirtymällä tarkoitetaan tässä työssä ajoneuvon siirtymistä digitaalisen  
kartan segmentiltä toiselle sellaisessa kohdassa, jossa segmentit eivät tosiasiaa ole yhdis-  
tyviä. Vapaan tilan siirtymiä kutsutaan jatkossa lyhenteellä VTS. VTS:n vieminen HMM:iin  
vaatii neljän ehdon täyttymisen: siirtymän on alun perin oltava hyvin epätodennäköinen  
(pitkä koukkaus, jonka pituus ylittää reilusti euklidisen etäisyyden seuraavalle pisteelle),  
siirtymän on tapahduttava katuverkon ja kevyen liikenteen väylän tai ulkoilureitin välillä,  
ajoneuvon nopeuden on oltava pieni ja VTS:n on oltava lyhyt. Jos ehdot täyttyvät, reittikan-



didaatin geometria palautetaan alkuperäisen lähde-kohde-parin muodostamaksi suoraksi viivaksi ja sen  $\Delta_{path}$ -arvo pakotetaan sopivaksi. Sopiva arvo on sellainen, joka saa transitiotodennäköisyyden, mikä voittaa helposti intuitiivisesti huonot vaihtoehdot, mutta häviää jonkin verran realistisille siirtymävaihtoehdoille. Kuvassa 16 on kuvattu työkoneen siirtymä kadulta ulkoilureitille ensin siten, että VTS:t eivät ole sallittuja ja sitten niiden kanssa. Reititivalinnat kuvan kahden sovitusvaihtoehdon välillä ovat identtiset lukuun ottamatta pisteiden 16-18 välisiä siirtymiä. Kuvasta havaitaan, että mikäli karttasovitus on tiukasti tieverkkoon rajattu, tulee todennäköisimmäksi reitiksi vaihtoehto, jossa kuljetaan katua pitkin pisteeseen 17 kohdalle asti, jonka jälkeen joudutaan palaamaan aina pisteelle 15 asti edelliseen risteykseen. Vasta sitä kautta voidaan tehdä siirtymä pisteelle 18 kevyen liikenteen väylää pitkin. Kun VTS:t ovat sallittuja, muodostuu sellainen pisteiden 16 ja 17 välille johtuen siirtymän todella suuresta  $\Delta_{path}$ -arvosta, ja todennäköisin reitti kulkee sujuvasti VTS:n kautta ulkoilureitille.



Kuva 16: Vapaan tilan siirtymät karttasovituksessa.

## 7.2 Sumea karttasovitus

Sumean karttasovituksen yhtenä hyötynä on erilaisten heterogeenisten parametrien joustava lisääminen malliin ja niiden välisten suhteiden määrittely intuitiivisesti. Kun HMM-algoritmissa käytettiin ainoastaan havaintopisteen etäisyyttä tieverkkoon kuvaamaan emissiotodennäköisyyttä ja reittikandidaattien pituuksien ja havaintopisteiden euklidisten etäisyyksien eroja kuvaamaan transitiotodennäköisyyttä, pyrittiin kehitetyssä sumeassa karttasovitusalgoritmissa ottamaan huomioon lisäksi kulkusuuntaan, aikaan ja nopeuteen liittyviä parametreja.

Algoritmissa hyödynnettiin seuraavia parametreja:

1)  $d$ , segmenttikandidaatin euklidinen etäisyys havaintopisteestä metreinä.

2)  $\Delta h$ , segmenttikandidaatin orientaation ero ajoneuvon kulkusuuntaan asteina. Tässä parametrissa verrataan ajoneuvon havaintopisteen kulkusuuntaa niiden segmenttien orientaatioon, joille pistekandidaatit on projisoitu. Orientaatio ja kulkusuunta normalisoidaan siten, että suuntaero  $\Delta h$  asettuu asteissa välille  $[0.0, 90.0]$  eli segmenttien digitointi- tai ajosuuntaa ei oteta huomioon. Paras arvo parametrille seuraa siitä, että kulkusuunta on yhdensuuntainen segmentin kanssa riippumatta siitä, onko se segmentin alusta loppuun vai toisin päin, ja huonoin arvo siitä, että kulkusuunta on kohtisuora segmenttiin nähden. Satelliittipaikannuksen perusteella määritelty kulkusuunta on hyvin epäluotettava, mikäli ajoneuvon nopeus on liian pieni. Ochieng et al. (2004) tutki nopeuden vaikutusta kulkusuunnan määrittämisen virheeseen kenttäkokein, joiden mukaan alle 3 m/s nopeudessa virhe kasvaa hyvin suureksi. Tästä johtuen sumea järjestelmä ei huomioi tätä parametria, mikäli ajoneuvon nopeus on alle 11 km/h (eli 3,06 m/s).

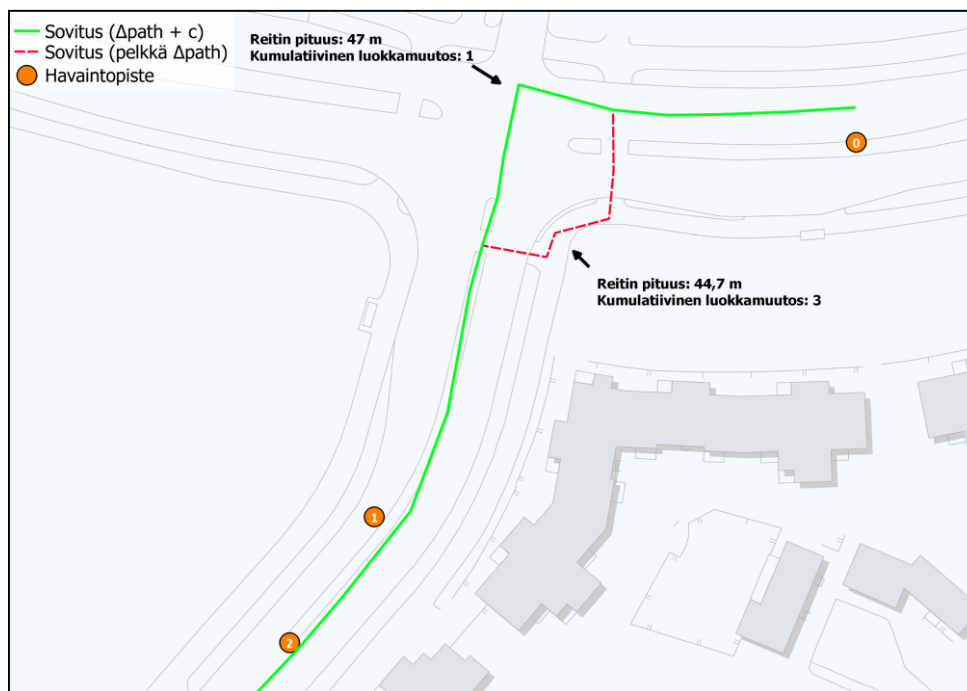
3)  $vs_p$ , pistemäisen nopeushavainnon suhde segmenttikandidaatin nopeusrajoitukseen. Tässä parametrissa verrataan havaintopisteellä mitattua ajoneuvon nopeutta pistekandidaatin segmentin nopeusrajoitukseen. Arvo 1,0 ja sitä suuremmat arvot tarkoittavat, että nopeus on pienempi kuin nopeusrajoitus, ja pienemmät tarkoittavat, että nopeus on suurempi kuin nopeusrajoitus. Koska tässä tutkimuksessa karttasovituksen sovelluksena oli katujen kunnossapito, ei nopeusrajoitusta pienemmästä nopeudesta haluttu rangaista, koska se on kunnossapitoa suorittaville ajoneuvoille tyypillistä ajokäyttäytymistä (Tiittanen 2016). Tavallisen tieliikenteen sovelluksissa voitaisiin hyvin suosia sellaisia reittejä, joiden nopeusrajoitus sopii mahdollisimman hyvin ajoneuvon nopeuteen.

4)  $\Delta_{path}$ , havaintopisteiden välisen euklidisen etäisyyden ja pistekandidaattien välisen tieverkkoa pitkin kulkevan lyhyimmän reitin pituuksien suhdeluku. Tämä on sama muuttuja kuin HMM:n transiitodennäköisyys, eli reitin pituuden ja pistekandidaattien välisen euklidisen etäisyyden suhteellinen ero. Sumeassa järjestelmässä jakauma on kuitenkin mallinnettu siten, että arvo 0 tuottaa suurimman todennäköisyyden, eli se määritellään pisteiden välisten etäisyyksien erosta kaavalla  $abs(1 - d / d_{path})$ .

5)  $vs_r$ , reittikandidaatin nopeusvaatimuksen suhde reitin keskimääräiseen nopeusrajoitukseen. Reittikandidaattien laskennan jälkeen jokaisen reitin segmenttien pituudet laskettiin yhteen, josta saatiin selville nopeus, jolla kyseinen reitti olisi pitänyt kulkea ottaen huomioon reitin alkupisteen aikaleiman  $t_i$  ja loppupisteen aikaleiman  $t_{i+1}$  erotuksen eli reitillä käytetyn ajan. Koska Digiroadissa ei ole katuverkolla kattavaa nopeusrajoitustietoa, käytettiin segmenttien nopeusrajoituksina toiminnallisesta luokasta pääteltyä nopeusarvioita. Kevyen liikenteen väylien nopeusrajoituksena käytettiin arvoa 28 km/h, sillä työkonet eivät yleensä liiku kevyen liikenteen väylillä yli 25 km/h nopeudella (Tiittanen 2016). Segmenttien pituuksista ja nopeusrajoitusarvioista saatiin laskettua keskimääräinen nopeusrajoitusten mukainen nopeus. Reittikandidaatin nopeusvaatimuksen ja nopeusrajoitusarvioista saadun järjestyksen kulkunopeuden suhde otettiin parametriksi sumeaan järjestelmään. Arvo 1,0 ja sitä suuremmat tarkoittavat sitä, että reittikandidaatti on nopeusvaatimuksensa suhteen järjestyksen mukainen, kun taas esimerkiksi 0,5 tarkoittaa sitä, että nopeusvaatimus on kaksi kertaa suurempi tai puolet pienempi kuin keskimääräinen nopeusrajoitusarvio. Työkoneiden tapauksessa voitai-

siin käyttää reitin keskimääräisen nopeusrajoituksen sijaan miniminopeusrajoitusta, sillä koneet liikkuvat usein kadullakin hitaammin, kuin tien nopeusrajoitus sallii. Miniminopeusrajoituksen hakeminen reitille voisi auttaa erottamaan reitit, joissa käytetään kevyen liikenteen väyliä verrattuna kadulla ajamiseen. Miniminopeusrajoituksen käyttämisestä seuraa se, että mikä tahansa reittikandidaatti, joka kulkee esimerkiksi kevyen liikenteen väylän kautta saa huonon arvon tästä parametrasta, mikäli sen nopeusvaatimus on merkittävästi suurempi kuin 25 km/h. Parametrien empiirisessä testauksessa menetelmien välillä ei kuitenkaan havaittu suurta eroa, ja lopullisessa algoritmissa käytetään keskimääräistä nopeusrajoitusta. Sovel-luksen kannalta parhaiden nopeusrajoitusten ja minimi- ja keskiarvomenetelmien vaikutus-ten tarkempi arviointi jää tehtäväksi tulevaisuudessa.

6) *c*, reittikandidaatin kumulatiivinen toiminnallisen luokan muutos. Samalla, kun reitin no-peusvaatimus laskettiin, voitiin selvittää myös reittikandidaattien toiminnallisen luokan ku-mulatiivinen muutos. Toiminnallisille luokille annettiin numeerinen luokkaa kuvaava koko-naislukuarvo 0-6, jossa arvon 0 saivat ulkoilureitit ja kevyen liikenteen väylät, ja arvon 6 valtatie. Aina kun toiminnallinen luokka muuttuu reittikandidaatin aikana, lisätään luokka-muutosten arvoon  $|\text{edellisen segmentin luokka} - \text{nykyisen segmentin luokka}|$ . Pienemmät ar-vot tässä parametrissa tekevät reitistä todennäköisemmän kuin suuremmat. Tämä parametri nojaa intuitioon, jonka mukaan edestakaiset muutokset tieluokkien välillä ovat harvinaisia, ja ajoneuvot pyrkivät kulkemaan tietyn tieluokan reittiä pitkin ja pysyttelemään sellaisella. Myös erittäin suuret luokkavaihtelut ovat epätodennäköisiä, esimerkiksi siirtymä kevyen lii-kenteen väylältä valtatielle. Tämä parametri lisättiin sumeaa järjestelmään kontrolloimaan sovituksessa havaittua kevyen liikenteen väyliä ja tieverkon välistä epäintuitiivista vaihte-lua. Pelkkää  $\Delta_{\text{path}}$ -parametria käytettäessä risteyksissä, joissa liitytään esimerkiksi liityntä-kadulta kokoojakadulle, jonka vieressä kulkee kevyen liikenteen väylä, algoritmi pyrkii oi-kaisemaan valitsemalla reitin liityntäkatu-KLV-kokoojakatu, sillä se voi tällaista reittiä käyt-tämällä minimoida  $\Delta_{\text{path}}$ -parametrin arvon. Kuvassa 17 on esimerkki tällaisesta tapauksesta. Toiminnallisen luokan turhasta vaihtelusta rankaisemalla tällaisia virheellisiä sovituksia voi-daan vähentää.



Kuva 17: Parametrin *c* (kumulatiivinen luokkamuu-tos) vaikutus työkonien risteyssovitukseen.

Edellä kuvatut parametrit jaettiin kahteen osaan, joista 1-3 kuvaavat pistekandidaatin todennäköisyyttä olla oikealla reitillä, ja 4-6 kuvaavat siirtymän todennäköisyyttä pistekandidaatin välillä. Niitä voi hyvin verrata HMM-algoritmin emissio- ja transitiotodennäköisyyksiin.

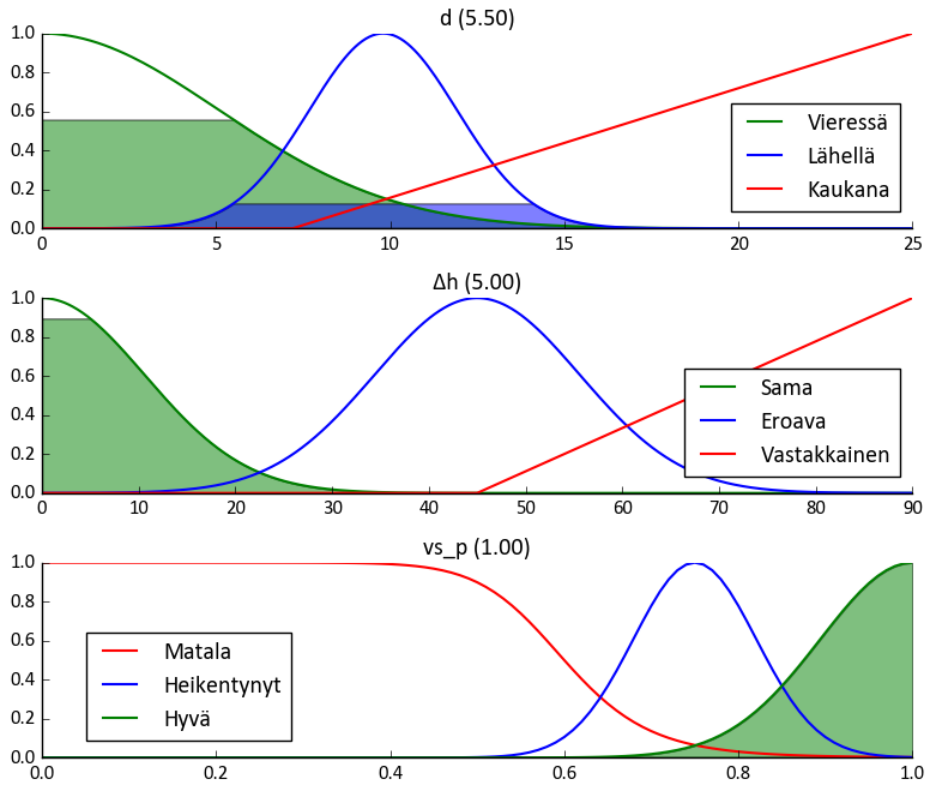
Sumean järjestelmän säännöt ovat seuraavat:

1. *JOS*  $d = \text{vieressä}$  *NIIN*  $p_e = \text{suuri}$
2. *JOS*  $d = \text{lähellä}$  *TAI*  $\Delta h = \text{eroava}$  *TAI*  $vs_p = \text{heikentynyt}$  *NIIN*  $p_e = \text{keskisuuri}$
3. *JOS*  $d = \text{kaukana}$ , *TAI*  $\Delta h = \text{vastakkainen}$  *TAI*  $vs_p = \text{matala}$  *NIIN*  $p_e = \text{pieni}$
4. *JOS*  $\Delta_{path} = \text{sama}$  *NIIN*  $p_t = \text{suuri}$
5. *JOS*  $\Delta_{path} = \text{pieni}$  *ero* *TAI*  $vs_t = \text{heikentynyt}$  *NIIN*  $p_t = \text{keskisuuri}$
6. *JOS*  $\Delta_{path} = \text{suuri}$  *ero* *TAI*  $vs_t = \text{matala}$  *TAI*  $c = \text{paljon}$  *NIIN*  $p_t = \text{pieni}$

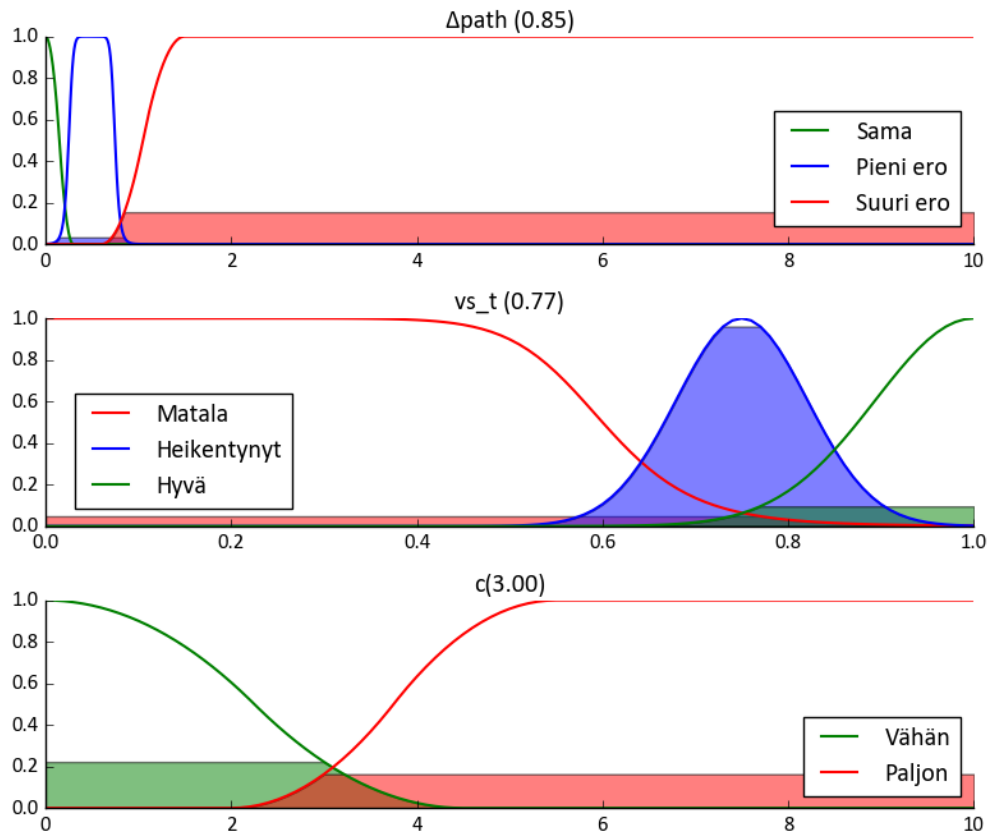
Lopullinen sumea todennäköisyys lasketaan sumean järjestelmän tulosjäsenyysfunktioiden massakeskipisteen perusteella selkeytettyjen arvojen tulona:  $p = p_e p_t$ .

Algoritmin alkuvaihe toteutettiin samalla tavalla kuin HMM-algoritmissakin. Pisteet luettiin tietokannasta, jonka jälkeen niiden perusteella haettiin tarpeeksi suuri osa tieverkkoa laskenta varten. Pisteille haettiin pistekandidaatit tieverkosta lähimmän naapurin haulla, ja näistä pisteistä muodostettiin lähde-kohde-parit. Pareille etsittiin tieverkosta lyhyin reitti, eli reittikandidaatti. Reittikandidaatit toimivat viimein syöteinä sumealle karttasovitusalgoritmillemme. Karttasovitusalgoritmi toteutettiin samaan tapaan Pythonilla kirjoitettuna kustomoituna muuntimena kuin HMM-algoritmikin. Kirjastot, joita algoritmista käytettiin, olivat Numpy, Matplotlib sekä scikit-fuzzy, joka on SciPy:n sumean logiikan kirjasto.

Reittikandidaatti ja sille lasketut attribuutit kuvaavat yhtäaikaaisesti sen lähtöpistettä koskevia emissiomuuttujia, ja seuraavalle pistekandidaatille vaadittavan siirtymän transitiomuuttujia. Sumeassa järjestelmässä lasketaan erikseen jokaisen reittikandidaatin emissiotodennäköisyys ja transitiotodennäköisyys. Emissiotodennäköisyyttä kuvaavat jäsenyysfunktiot on esitetty kuvassa 18 ja transitiotodennäköisyyttä kuvaavat jäsenyysfunktiot kuvassa 19. Tulosten aggregointi tehdään maksimimenetelmällä ja selkeytys massakeskipistemenetelmällä. Tulostulosjäsenyysfunktioiden muodot on esitetty kuvassa 20.

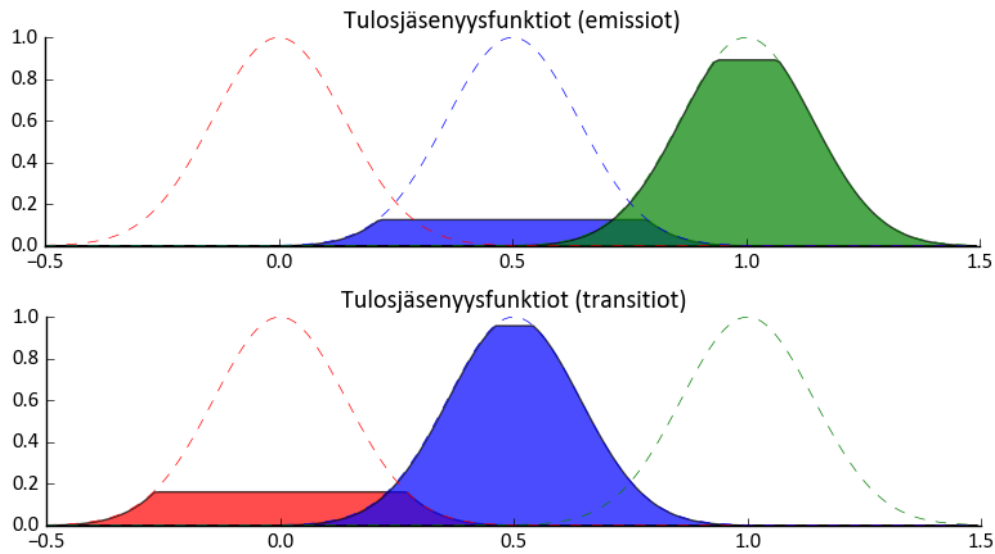


Kuva 18: Emissiomuuttujien jäsenyysfunktioiden muodot ja aktivoituminen syötteellä  $d=5.5$ ,  $\Delta h=5.0$ ,  $vs_p = 1.0$ .



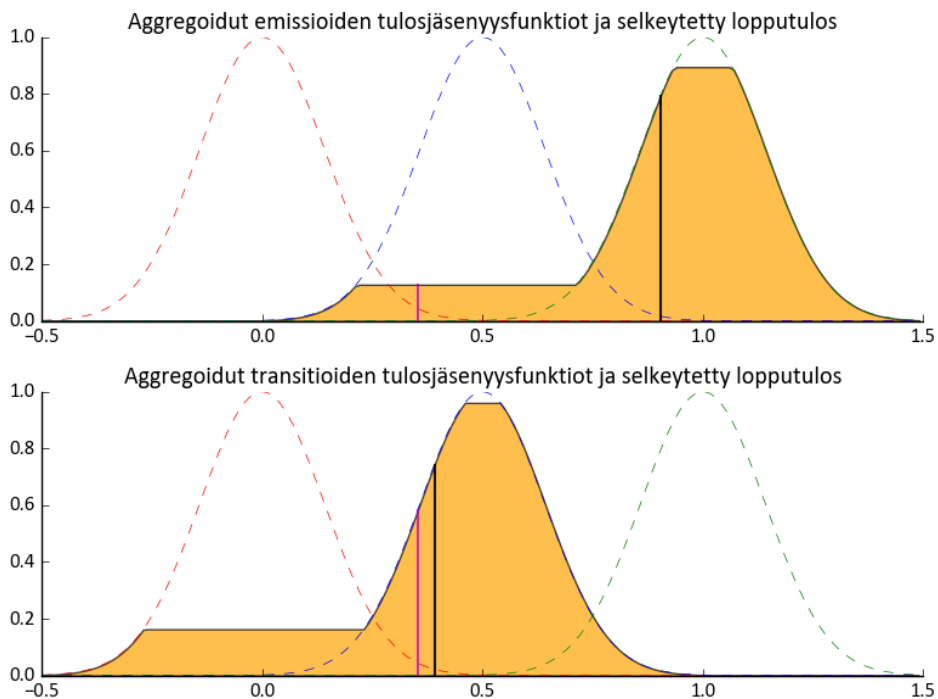
Kuva 19: Transitiomuuttujien jäsenyysfunktio ja aktivoituminen syötteellä  $\Delta_{path}=0.85$ ,  $vs_t=0.77$ ,  $c=3$ .





Kuva 20: Tulosjäsenyysfunktioiden aktivoituminen sääntöjen evaluoinnin jälkeen.

Kun tulosjäsenyysfunktioiden aktivoituminen on saatu selville, kaikki sumeat tulosjoukot  $\{Matala, Neutraali, Korkea\}$  aggregoidaan yhteen maksimimenetelmällä. Selkeytys tehdään laskemalla aggregaattien massakeskipisteet. Täten saadaan kaksi erillistä tulosta, transitioille ja emissioille. Lopullinen tulos  $p$  lasketaan näiden arvojen tulona. Kuvassa 21 esitetään aggregoidut tulosjäsenyysfunktiot ja erillisten selkeytysten tulosarvot sekä lopullinen tulos.

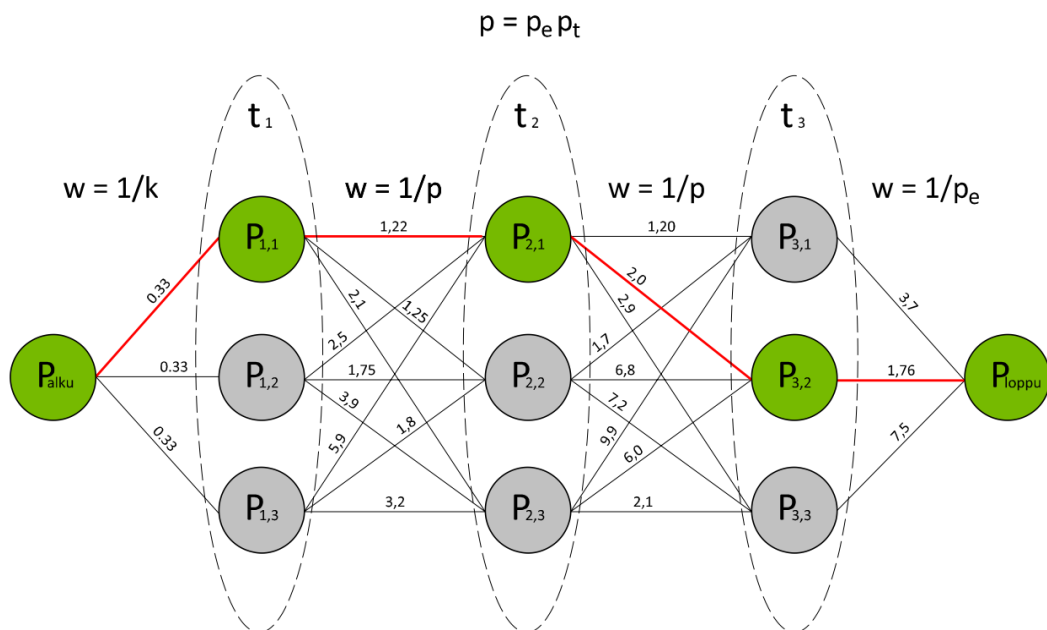


Kuva 21: Aggregoidut tulosjäsenyysfunktiot ja selkeytetyt lopputulokset merkittynä mustilla viivoilla. Selkeytysten tulo on merkitty molempiin kuvaajiin violetilla viivalla ( $0,9 \cdot 0,39 = 0,35$ ).

On huomattava, että tulosavaruuden x-akseli ei ole normaalin todennäköisyyslaskennan  $[0.0, 1.0]$ , vaan sitä on venytetty siten, että maksimi- ja miniarvot, jotka yksittäinen syöte voi saada tulokseksi, asettuu välille  $[0.0, 1.0]$ . Massakeskipistemenetelmää käytettäessä selkeytyksen ääriarvot ovat tulosjäsenyysfunktioiden massakeskipisteitä. Tästä johtuen esimerkiksi gaussisen jäsenyysfunktion keskiarvon on oltava ääriarvon kohdalla (tässä tapauksessa 0,0 tai 1,0), ja tulosavaruuden x-akselin on oltava niin suuri, että jäsenyysfunktiot mahtuvat halutulla keskihajonnalla kokonaisuudessaan sille.

Kun sumea järjestelmä on evaluoitu ja sumea todennäköisyys  $p$  on saatu selville, muodostetaan painotettu kandidaattigraafi hieman samaan tapaan kuten Lou et al. (2009). Kandidaattigraafiin lisätään solmuiksi kaikki pistekandidaatit ja niiden välille lisätään kaaret siten, että pisteen  $p_i$  pistekandidaateista  $c_{i,k}$  on yhteydet kaikkiin pisteen  $p_{i+1}$  pistekandidaatteihin  $c_{i+1,k}$ . Kaarien painoiksi tulee kyseistä yhteyttä kuvaavan reittikandidaatin sumean todennäköisyyden  $p$  käänteisluku  $1/p$ . Lisäksi graafiin lisätään virtuaalinen aloitusolmu, josta on yhteydet kaikkiin  $p_1$  pistekandidaatteihin painolla  $1/k$ , ja virtuaalinen lopetusolmu, johon on yhteydet kaikista viimeisen pisteen  $p_n$  solmuista käyttäen painona ainoastaan viimeisen pisteen emisioimuuttujien sumean todennäköisyyden käänteislukua  $1/p_e$ .

Globaali optimireitti haetaan graafista etsimällä lyhyin reitti Dijkstran algoritmilla aloitusolmusta lopetusolmuun. Lopputuloksena algoritmi antaa sumean todennäköisyyden maksimoivan reittikandidaattien sekvenssin, jonka perusteella tietorakenteesta haetaan näitä reittejä vastaavat kohteet ja tulostetaan ne. Kuvassa 22 on havainnollistettu kandidaattigraafin rakennetta, kun naapureiden maksimilukumäärä  $k$  on kolme ja havaittuja pisteitä on kolme.



Kuva 22: Reitin haku kandidaattigraafista.  $P_{i,k}$  kuvaa ajan  $t_i$  havaintopisteen eri pistekandidaatteja, joiden välille on määritetty mahdollisia siirtymiä kuvaavat kaaret sumean järjestelmän laskemilla painoilla.

Sumeaa karttasovitususta testattiin samalla aineistolla kuin HMM-algoritmiakin (luku 7.1.3), ja sille toteutettiin samat viritykset (luku 7.1.4) eli:

1. Liikenneverkon suunnan mallinnus lyhyimmän reitin haussa,
2. lähekkäin olevien pisteiden sulautus yhteen,
3. liikenneverkon jakaminen kahdeksi erilliseksi graafiksi, joista vain toisessa on mukana kevyen liikenteen väylät,
4. havaintopisteiden poistaminen tunnetuilta tieverkon ulkopuolisilta alueilta,
5. vapaan tilan transitioiden salliminen ja
6. yli 40 m/s nopeuden vaativien reittikandidaattien poistaminen laskennasta.

Parametrien estimointivaiheessa havaittiin, että työkoneet vaativat suuremman määrän lähimpiä naapureita pistekandidaattien haussa ( $k$ -parametri) kuin autot. Tämä selittyy luonnollisesti sillä, että tieverkko on tiheämpi, joten oikeaa reittiä vastaa aina suurempi määrä väärä reittejä. Pienemmillä  $k$ :n arvoilla havaittiin, ettei pistekandidaattien joukkoon yksinkertaisesti kuulunut ajoneuvon oikealle reitille osuvia pisteitä. Saman havaittiin toistuvan myös suuremmilla  $k$ -määrillä, vaikkakin harvemmin, erityisesti sellaisissa kohdissa, joissa lähimmän naapurin haussa saatiin tulokseksi suuri määrä kevyen liikenteen väyliä tai ulkoi-lureittejä, mutta esimerkiksi nopeuden järkeyystarkistus pystyi alentamaan näiden reittien todennäköisyyksiä. Tämän havainnon johdosta segmenttikandidaattien hakuun tehtiin muutos, joka takaa sen, että vähintään puolet pistekandidaateista sijoittuu tavalliselle katuverkolle ja loput läheisyysjärjestyksessä joko katuverkolle tai kevyen liikenteen verkolle.

### 7.3 Algoritmien vertailu

Karttasovitusalgoritmien sovitustarkkuutta on kirjallisuudessa vertailtu kaikkein yleisimmin tutkimalla, kuinka moni piste on sovitettu oikealle tieverkon segmentille, kun sovitustulosta verrataan referenssiaineistoon (esimerkiksi Quddus et al. 2006, Kim et al. 2001 ja Velaga et al. 2009). Sovelluksissa, joilla on pyritty parantamaan satelliittipaikannuksen tarkkuutta ajonaikaisessa käytössä, on käytetty usein mittarina segmentinsuuntaista virhettä ja segmentinvastaista virhettä (esimerkiksi Kim et al. 2001 ja Quddus 2006). Segmentinsuuntainen virhe tarkoittaa paikannuksen tarkkuutta segmentillä, ja segmentin vastainen virhe on käytännössä seurausta digitaalisen kartan keskilinjavirheestä. Näiden yhdistelmänä saadaan sovitettujen pisteiden keskimääräinen horisontaalinen virhe, jota on myös käytetty mittarina arvioitaessa karttasovituksen tulosta (esimerkiksi Quddus ja Ochieng 2014). Keskimääräisen sovitusvirheen mittaaminen vertaamalla sovitetun havaintopisteen etäisyyttä ajoneuvon todelliseen sijaintiin edellyttää käytännössä referenssiaineiston keräämistä jollakin erittäin tarkalla paikannusmenetelmällä, esimerkiksi integroidulla RTK/DR-paikannuksella.

Sellaisia karttasovitusmenetelmiä testattaessa, jotka sovittavat koko ajoneuvon kulkeman reitin kerralla, on mittarina käytetty usein väärin sovitetun reitin osuutta todellisesta reitistä. Tällainen testaustapa ottaa huomioon sekä karttasovitukselta puuttuvat osuudet todellisesta reitistä, että karttasovituksen tuottamat ylimääräiset osuudet, joita todellisella reitillä ei kuljettu. Tällaista mittaria ovat käyttäneet esimerkiksi Newson ja Krumm (2009), Osogami ja Raymond (2013) ja Koller et al. (2015).

Kaikki edellä mainitut mittarit nojaavat jonkinlaisen referenssiaineiston keräämiseen. Algoritmien testausmenetelmiä, joissa ei käytetä lainkaan referenssiaineistoa, ei kirjallisuudessa

esiinny kovin paljon. Szwed ja Pekala (2013) käyttivät esimerkiksi HMM-algoritminsa suorituskyvyn arviointiin HMM:n uudelleenalustusten lukumäärää eli liikeradan katkaisujen lukumäärää. Algoritmi arvioitiin paremmaksi pienemmällä uudelleenalustusten määrällä. Tällaisten epäsuorien mittareiden kytkös todellisten sovellusten sovitustarkkuusvaatimukseen on kuitenkin heikko. Epäsuoria menetelmiä voikin käyttää korkeintaan yksittäisen algoritmin parametrien estimointiin.

Referenssiaineiston keräämiseen karttasovituksen testaamiseksi kirjallisuudessa on käytetty seuraavia strategioita:

1. *Reittien manuaalinen sovitus tieverkkoon.* Data on kerätty ajoneuvoista, joissa on paikannuslaite, ja data sovitetaan manuaalisesti tieverkkoon ihmisen päättelykykyä käyttäen. Menetelmä toimii yleensä hyvin, mutta paikannusdata saattaa olla niin harvaa tai niin epävarmaa, ettei ihminen osaa päätellä datan perusteella kuljettua reittiä. Strategiaa ovat käyttäneet esimerkiksi Lou et al. (2009), Li et al. (2013), Yang ja Meng (2014) ja Xu et al. (2014).
2. *Tarkempi paikannusmenetelmä.* Data on kerätty ajoneuvoista, joissa on karttasovituksessa käytettävää dataa tuottava paikannuslaite ja lisäksi niin tarkkaa paikannusdataa tuottava paikannuslaite, että esimerkiksi sen perusteella tehty lähimmän naapurin haku tuottaa käytännössä aina oikean tuloksen. Yleisiä yhdistelmiä on esimerkiksi itsenäinen satelliittipaikannin ja RTK-mittausta hyödyntävä tarkka satelliittipaikannin tai integroitu RTK/DR-paikannusjärjestelmä. Tällaisen datan avulla voidaan myös tehdä manuaalinen sovitus, jolloin kaikki manuaaliseen sovittamiseen liittyvä epävarmuus voidaan poistaa. Tällaista strategiaa ovat käyttäneet esimerkiksi Quddus (2006) ja Quddus ja Ochieng (2014).
3. *Itse kerätty data.* Karttasovitettua dataa tuottavalla paikannuslaitteella varustetulla ajoneuvolla ajetaan testireittejä, jotka suunnitellaan etukäteen. Strategiaa ovat käyttäneet esimerkiksi Kim ja Kim (2001) ja Newson ja Krumm (2009).
4. *Tunnetut reitit.* Referenssiaineistoa voidaan kerätä tunnettuja reittejä pitkin liikkuvista ajoneuvoista, jotka tuottavat karttasovituksessa käytettävää paikannusdataa. Tällaisia voivat olla esimerkiksi tunnettuja bussireittejä pitkin ajavat bussit. Strategiaa ovat käyttäneet esimerkiksi Goh et al. (2012).
5. *Simuloitu data.* Jos referenssiaineistoa ei ole saatavissa, sitä voidaan luoda simuloimalla tieverkosta olettamalla tietty keskilinjavirhe ja paikannusvirhe sekä havaintoväli. Liikennevirtoja ja etenkin kuljettajien käyttäytymistä on kuitenkin vaikeaa simuloida, eikä tällaisella datalla tehtyjen testien tuloksia voikaan pitää täysin luotettavina. Esimerkiksi algoritmin kehityksenaikaiseen testaukseen strategia voi kuitenkin sopia. Tätä strategiaa ovat käyttäneet esimerkiksi Krakiwsky et al. (1988) ja Jagadeesh et al. (2004).

Tässä tutkimuksessa algoritmien sovitustarkkuuden vertailu tehtiin käyttämällä muunneltua Newsonin ja Krummin (2009) reittien sovitusvirheestä, sillä luotavan sovelluksen keskiössä on ajoneuvojen kulkemien reittien analysointi. Sillä ei ole merkitystä, kuinka tarkasti

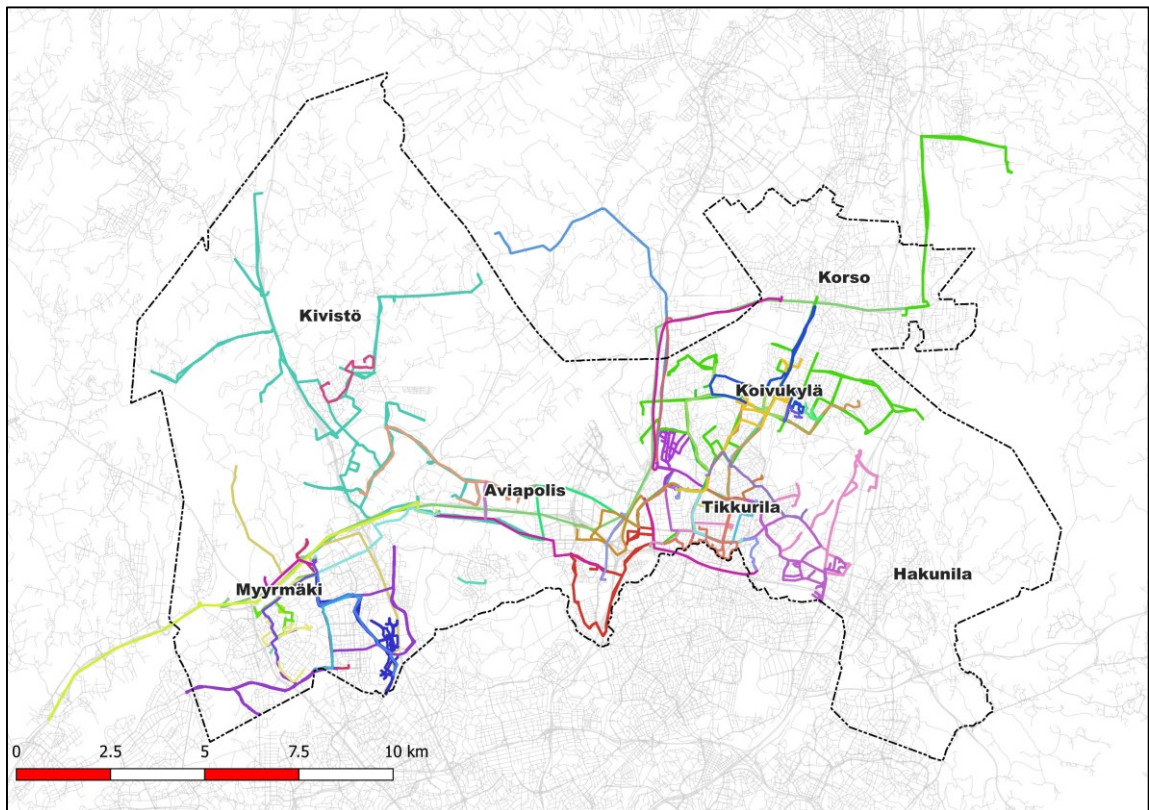
yksittäinen havainto sijoittuu kuljetulle reitille, tai edes sillä, sijoittuuko havainto juuri oikealle segmentille, kunhan se sijoittuu reitille, jonka ajoneuvo todellisuudessa on kulkenut. Mittaria muunneltiin siten, että saadaan intuitiivisempi reitin sovitustarkkuus %, joka saadaan kaavalla

$$\left(1 - \frac{d_c + d_o}{d_r}\right) \times 100 \% \quad (7)$$

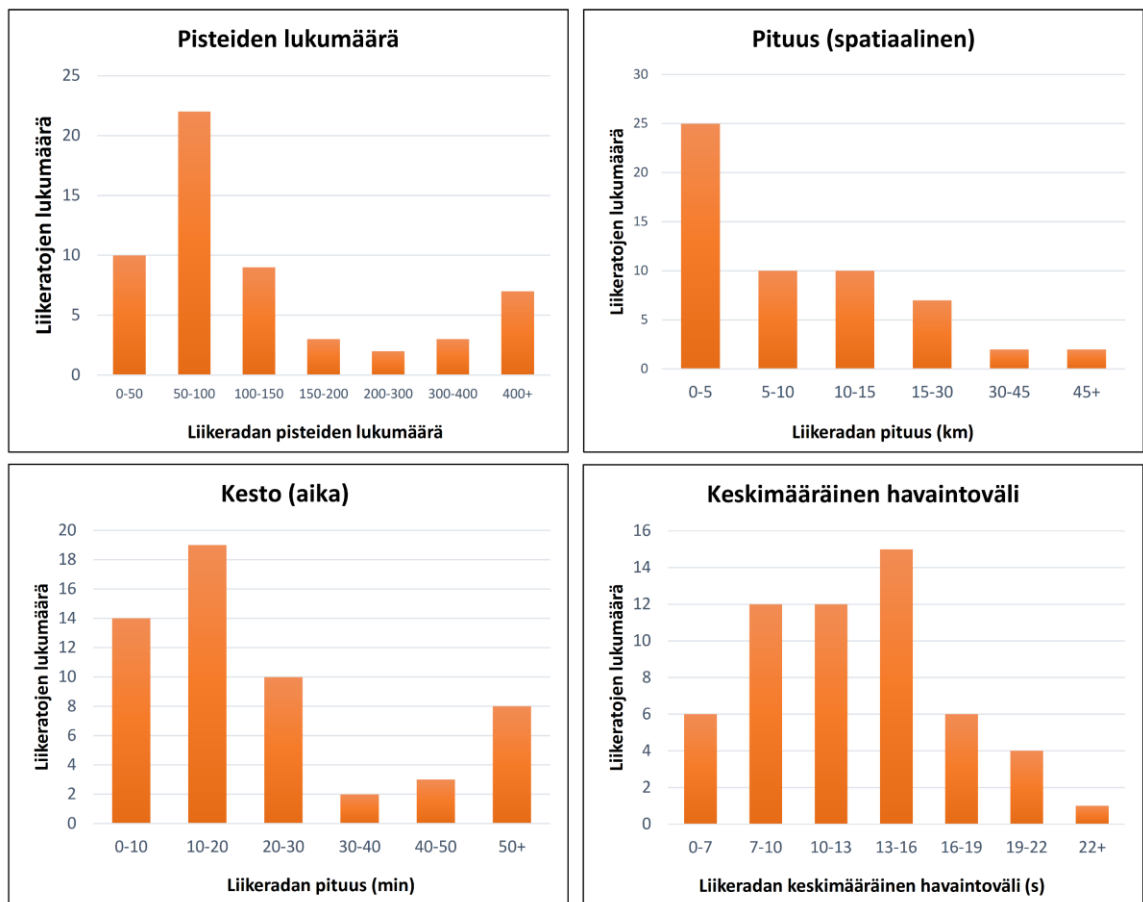
missä  $d_c$  tarkoittaa sovitusalgoritmin tuottamien ylimääräisten reittiosuuksien pituuksia,  $d_o$  tarkoittaa todellisten reittiosuuksien pituuksia, joita algoritmi ei tuottanut ja  $d_r$  tarkoittaa todellisen reitin pituutta. Referenssiaineisto puolestaan luotiin keräämällä havaintotietokannasta kokonaisia liikeratoja ja sovittamalla ne manuaalisesti tieverkkoon. Algoritmien kehittämistä ja testausta varten luotiin pienempi referenssiaineisto *testaus* (luku 7.1.3). Seuraavassa luvussa esitetään algoritmien lopullisen vertailun tulokset käyttäen laajempaa referenssiaineistoa *validointi*.

### 7.3.1 Referenssiaineisto

Validointiaineisto kerättiin poimimalla satunnaisotannalla liikeratoja havaintopisteiden tietokannasta samoilla säännöillä kuin luvun 7.1.3 testausaineisto. Liikeradan pituuden tuli olla vähintään 750 metriä ja koostua ainakin 20 havaintopisteestä siten, että peräkkäisten havaintojen aikaero sai olla korkeintaan 60 sekuntia. Lopullinen referenssiaineisto koostui 56 liikeradasta, joista työkoneiden tuottamia oli 28 ja autojen tuottamia 28. Liikeradat koostuivat yhteensä 10 052 pisteestä, joiden kokonaispituus oli 676 km ja kesto tunteina 27 h. Kuvassa 23 on esitetty referenssiaineisto kartalla ja kuvassa 24 histogrammit liikeratojen a) pituudesta metreinä, b) pisteiden lukumäärästä, c) pituudesta minuutteina ja d) keskimääräisestä havaintovälistä.



Kuva 23: Referenssiaineisto esitettyä tieverkolla. Erillisillä liikeroadoilla on uniikit värit.



Kuva 24: Referenssiaineiston ominaisuudet.

### 7.3.2 Vertailun tulokset

Algoritmeja vertailtaessa autot (kuorma-autot ja henkilöautot) ja työkoneet (pyöräkuormaajat ja traktorit) erotettiin toisistaan. Erotus tehtiin, koska näiden eri ajoneuvotyyppien liikeradat poikkeavat toisistaan huomattavasti, erityisesti johtuen siitä, että työkoneilla tehdään kevyen liikenteen väylien ja ulkoilureittien ylläpitoa. Karttasovitus on siten huomattavasti helpompaa toteuttaa autoille kuin työkoneille.

Vertailussa käytettiin samankaltaista FME:llä luotua vertailuprosessia kuin luvussa 7.1.3. Prosessissa manuaalisesti sovitettuja referenssireittejä verrattiin eri algoritmien havaintopisteistä tuottamiin lopputuloksiin ja näistä laskettiin sovitustarkkuus %. HMM- ja sumeiden reittien algoritmien lisäksi vertailuun otettiin mukaan yksinkertainen geometrinen piste-kaari-algoritmi. Algoritmi sovittaa havaintopisteet lähimmälle tieverkon segmentille ja hakee tieverkosta lyhyimmät reitit peräkkäisten pisteiden välille. Vertailun tulokset on esitetty taulukossa 5.

| <i>Algoritmi</i> | <i>Sovituksen tarkkuus:<br/>Autot</i> | <i>Sovituksen tarkkuus:<br/>Työkoneet</i> |
|------------------|---------------------------------------|---|
| Piste-kaari      | 94,3 %                                | 66,6 %                                    |
| HMM              | 98,5 %                                | 86,5 %                                    |
| Sumeat reitit    | <b>99,4 %</b>                         | <b>93,1 %</b>                             |

Taulukko 5: Algoritmien sovitustarkkuuden vertailutulokset.

Molempien algoritmien sovitustarkkuus autojen liikeratojen karttasovituksessa oli erinomainen. Etenkin sumeiden reittien algoritmilla sovituksen erot todellisen reitin kanssa johtuivat lähinnä pienistä eroista referenssireittien aloitus ja lopetuspisteissä suhteessa havaintopisteisiin sekä oikaisuista pysäkillisten bussikaistojen kautta Kehä III:llä, Tuusulanväylällä ja Hämeenlinnanväylällä. Bussikaistat minimoivat siirtymien  $\Delta_{path}$ -arvon aina, mikäli väylä kaartuu siten, että ne jäävät sisäkaarteeseen. Nopeusrajoitus ja luokkatiedot ovat kuitenkin aivan samat kuin varsinaisella valtatielläkin. Toisaalta tutkimuksen aiheena olleen sovelluksen johdosta tähän ongelmaan ei kiinnitetty paljoakaan huomiota, sillä nämä erityistapaukset esiintyvät aina sellaisilla tieosuuksilla, joiden kunnossapidosta vastaa Liikennevirasto eikä kaupunki. Tulevaisuudessa ongelma voitaisiin korjata jalostamalla käytettävää digitaalista tieverkkoa tiedolla tällaisista osuuksista, esimerkiksi alentamalla niiden nopeusrajoitusta tai yksinkertaisesti lisäämällä kustannusta (eli muodostetun graafin kaarien painoa).

Työkoneiden tapauksessa havaittiin suurempia eroja algoritmien sovitustarkkuudessa kuin autoilla. Sumeiden reittien algoritmi suoriutui tässä osiossa parhaiten, jopa melko suurella erolla HMM-algoritmiin, johtuen toisaalta nopeus- ja aikatietojen huomioon ottamisesta ja toisaalta siitä, että parametrien estimointi empiirisin testein vaikutti parantavan sovitustarkkuutta helpommin kuin HMM-parametrien vastaava estimointi. Sumeiden reittien algoritmin molemmissa osioissa saavuttamaa sovitustarkkuutta voidaan pitää erinomaisena, mutta työkoneiden osalta on tiedostettava, että mukana on myös väärin tehtyjä sovituksia. Nämä virheet tapahtuivat käytännössä aina sellaisissa kohdissa, joissa työkone liikkui joko pääkadun reunassa tai rinnakkain samaan suuntaan kulkevalla kevyen liikenteen väylällä alhaisella nopeudella. Paikannustarkkuus ei aivan riitä siihen, että voitaisiin yksiselitteisesti päätellä,

kummalla reitillä ajoneuvo on. Toisaalta, koska ajoneuvo liikkuu lähes suoraan, myös havaintoväli oli havaituissa ongelmakohdissa melko harva. Tällaisissa tapauksissa karttasovitus tehdään jompaa kumpaa epävarmaa reittiä pitkin, riippuen esimerkiksi siitä, miten hyvin digitaaliseen tieverkkoon mallinnetut siirtymät katuverkon ja kevyen liikenteen verkon välillä sattuvat osumaan havaintopisteisiin nähden.

Referenssiaineistoa kerätessä törmättiin tilanteisiin, joissa ajoneuvo selvästi vaihteli kevyen liikenteen väylän ja pääkadun välillä (mahdollisesti esimerkiksi pääkadun bussipysäkkien aurauksen tai hiekoituksen johdosta), mutta havaintoväli oli suhteellisen suoran liiketradan johdosta niin harva, ettei edes ihmisen päättelyllä ollut mahdollista sanoa, missä kohdassa ajoneuvo tarkalleen vaihtoi väylältä toiselle. Referenssiaineistoa kerätessä jouduttiin hylkäämään neljä liikerataa sen johdosta, että ne sisälsivät liikaa tällaisia reittivalintoja, joita ei pystytty määrittelemään tarkasti.

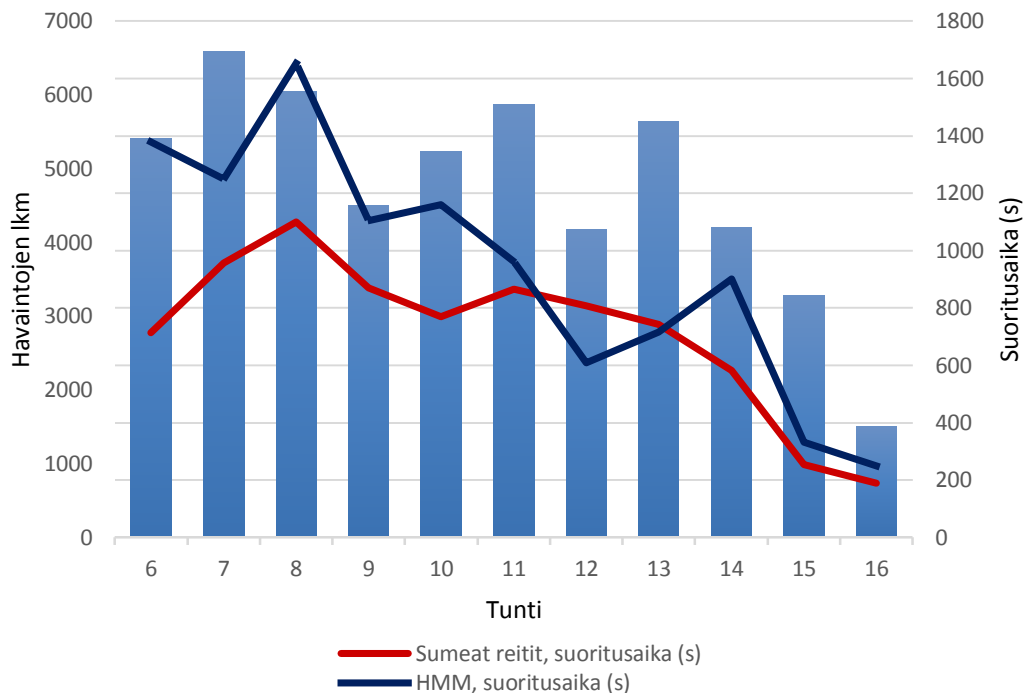
Käytetty vertailumenetelmä on toisaalta ankara mittari siinä mielessä, että väärin sovitettu reitti aiheuttaa tarkkuuden heikentymistä sekä puuttuvan reitin pituuden verran, että tehdyn ylimääräisen reitin pituuden verran. Toisin sanoen, mikäli puolet reittivalinnoista menee väärin, mittarissa saavutetaan noin 0 % sovitustarkkuus. Mikäli yli puolet reittivalinnoista on vääriä, sovitustarkkuudesta tulee negatiivinen. Toisaalta, koska vertaillaan kokonaisia reittejä, voi sovitustarkkuus nousta korkeaksi verrattuna pisteiden sovitustarkkuuteen esimerkiksi silloin, mikäli vertailtavat liikeradat sisältävät pitkiä siirtymiä harvan tieverkon tieosuuksilla, mikä tekee osuuskien oikein sovittamisesta helppoa.

### 7.3.3 Laskennallisen suorituskyvyn vertailu

Algoritmien laskennallisen suorituskyvyn vertailemiseksi havaintodatasta etsittiin kaikkein aktiivisin päivä vuoden 2016 talvelta. Koska havaintopisteitä haetaan tunnin välein, testattiin vertailtavien karttasovitusalgoritmien suorittamista päivän datalle siten, että yhteen ajoon valittiin aina tunnin data kerrallaan. Vertailupäiväksi valikoitui sunnuntai 21.2.2016, jolloin lumisateen jälkeen lähes koko aurauskalusto oli liikenteessä. Kalustoa käytettiin kello 5:00–16:00 välisenä aikana. Havaintopisteitä järjestelmään kertyi päivän aikana 53716, keskimäärin 4476 tunnissa. FME-prosessit, jotka suorittivat vertailtavat karttasovitusalgoritmit ja kirjoittivat tulokseksi saadut reitit tietokantaan, ajettiin päivän datalle tunti kerrallaan ja ajojen kestot laskettiin yhteen. Vertailun tulokset on esitetty kuvassa 25.



## Prosessin suoritusajat/tunti



Kuva 25: Algoritmien suoritusajat tunneittain talven aktiivisimpana päivänä (21.2.2016). Algoritmit on suoritettu karttasovittamalla tunnin havainnot kerrallaan. Tulos on kolmen suorituskerran keskiarvo.

HMM-karttasovituksen kokonaislaskenta-aika 11 tunnin datalle oli 10315 sekuntia ja sumeiden reittien algoritmin vastaavasti 7850 sekuntia. Karttasovitus tehdään kuitenkin jakamalla havaitut liikeradat sopiviin paloihin ja suorittamalla prosessi rinnakkaisajona. Testissä käytettiin maltillista kahden prosessin rinnakkaisajoa, minkä seurauksena prosessiin kului todellisuudessa aikaa HMM-algoritmeilla 4483 sekuntia ja sumeiden reittien algoritmeilla 3644 sekuntia. Kahden prosessin rinnakkaisajolla todellista aikaa käytettiin laskenta-aikaan verrattuna HMM:llä siis 43,5 % ja sumeilla reiteillä 46,4 %. Tarvittaessa rinnakkaisajoa voitaisiin tehdä esimerkiksi 4–8 rinnakkaisprosessilla. Data ryhmiteltiin rinnakkaisajoprosesseihin dynaamisesti hakemalla tietokannasta erillisessä ohjausprosessissa satunnaisotannalla korkeintaan 500 pistettä per ajoneuvo per tunti, laskemalla ajoneuvojen havaintopisteiden konveksipeitteet ja yhdistämällä lähekkäiset pistejoukot toisiinsa. Jokainen yhdistetty pistejoukko muodosti oman ryhmänsä, mikä vastasi tiettyjen ajoneuvojen joukkoa. Nämä ryhmät käsiteltiin erillisissä karttasovitusprosesseissa. Tällainen ryhmittely on erittäin hyödyllinen verrattuna esimerkiksi satunnaiseen ryhmittelyyn, sillä spatiaalisesti kompaktien liikeratajoukkojen käsittely on tehokasta, koska tällöin pystytään minimoimaan yksittäisessä sovitussuorituksessa tarvittavan tieverkon koko.

Sumeiden reittien algoritmi oli laskentaan käytettyä laskennallista suoritusaikaa vertailtaessa noin 31 % nopeampi kuin HMM-algoritmi, ja todellista käytettyä aikaa verrattaessa noin 23 % nopeampi. Minuuteissa ero tarkoittaa testatulla datalla 41 minuutin eroa laskenta-ajassa ja 14 minuutin eroa todellisessa käytetyssä ajassa. Algoritmeissa on havaittava nopeusero, mutta tutkimuksen sovelluksen suorituskyvyn kannalta se ei ole kovin merkittävä.

## 8 Karttasovituksen hyödyntäminen

Sumea karttasovitus algoritmi integroitiin osaksi prosessia, joka hakee uusia pisteitä rajapinnasta tunneittain. Kun uusia pisteitä on haettu tietokantaan, poimitaan ne karttasovitettavaksi ja tallennetaan tulos toiseen tietokantatauluun *sovitettut\_liikeradat*. Koska globaalit algoritmit pyrkivät ratkaisemaan kokonaisen liikeradan karttasovituksen kerralla, oli prosessissa otettava huomioon se, mitkä liikeradat ovat keskeneräisiä ja mitkä jo loppuneita.

Karttasovitetuille reiteille haettiin katuverkosta sen tien nimi, jota pitkin ne kulkevat, ja tarvittaessa reitit jaettiin useampaan osaan kadun vaihtuessa. Tämän jälkeen reitit yhdistettiin yhtenäisiksi lineaarisiksi kohteiksi siltä osin, kun niiden liikerataryhmä, statustieto ja kadun nimi olivat yhtenäisiä. Yhdistetyille reiteille laskettiin keskimääräinen aikaleima ja reitin nopeus reitin osien tietojen perusteella.

Ajoneuvojen tilatiedot tallennettiin omaan tauluunsa tietokantaan. Mikäli ajoneuvolla oli useampi aktiivinen tilatieto, muodostui näiden tilatietojen ja sovitettujen reitien välille monesta yhteen suhde. Attribuuteiksi sovitetuille reiteille tallennettiin aloitusaika, lopetusaika, ajoneuvon tunniste, tien nimi, pituus ja pituuden suhde kyseisen tieosuuden kokonaispituuteen. Uuden karttasovituskierron käynnistyessä keskeneräisiksi liikeradoiksi määritellään sellaiset tallennetut liikeradat, joiden lopetusaika on 60 s sisällä jostakin uudesta havaintopisteestä, jonka ajoneuvotunniste on sama kuin tallennetulla liikeradalla.

Sovitettujen liikeratojen taulua voidaan hyödyntää kyselyissä kohdistamalla hakuehtoja tien nimiin, aikaleimoihin, tilatietoihin, pituuteen ja geometriaan. Sen avulla on mahdollista vastata muun muassa seuraaviin kysymyksiin:

1. *Milloin viimeksi kadulla  $x$  on aurattu,*
2. *Milloin viimeksi katu  $x$  on aurattu kerralla kokonaan (jossa kokonaan voidaan määritellä esimerkiksi 80 % kokonaispituudesta),*
3. *Kuinka monena eri päivänä kadulla  $x$  on aurattu aikavälillä  $t_1 - t_2$ ,*
4. *Hae ajoneuvojen kaikki 200 m säteellä pisteestä  $p$  kulkemat reitit, jotka ovat syntyneet aikavälillä  $t_1 - t_2$ .*

Sovitettujen liikeratojen tallentamisen jälkeen samassa prosessissa muodostettiin lisäksi ajantasainen viimeisimmän tilannekuvan kertova tieto eri tilatiedoilla tehdyistä toimenpiteistä. Uusia karttasovitettuja liikeratoja verrattiin tämän tilannekuvataulun tietoihin ja sen tietoja päivitettiin siten, että jokaisesta mahdollisesta reitistä on tallennettuna viimeisin reittiä pitkin kulkeva karttasovitettu liikerata kullakin mahdollisella tilatiedolla. Tilannekuvataulua varten yksittäisten liikerataryhmien ja ajoneuvojen tiedot häivytettiin ja yhtenevät reitit yhdistettiin niiden sijaan statustiedon, päivämäärän ja kadun nimen perusteella. Yhdistetyille reiteille laskettiin vielä osareittien perusteella mediaaniaikaleima ja haettiin sen koneen tunnus, joka on aiheuttanut yhdistetyn reitin viimeisimmän havaintopisteen.

Prosessi tiedon hakemisesta karttasovitukseen ja ajantasaisen tilannetiedon tallentamiseen on kuvattu kaaviona liitteessä 3. Viimeisimmät tilannetiedot julkaistaan WMS- ja WFS-paikkatietorajapinnoissa. WMS eli Web Map Service ja WFS eli Web Feature Service ovat Open Geospatial Consortiumin (OGC) standardeihin perustuvia tapoja paikkatiedon (rasterimuotoisen kartan ja vektorimuotoisten kohteiden vastaavasti) jakeluun internet-yhteyden

välityksellä. OGC-rajapinnat toteutettiin GeoServerin kautta. Tämän lisäksi kaikkien sovitettujen reittien hakemiseksi ja niihin kohdistuvien kyselyiden tekemiseksi toteutettiin REST-tyylinen rajapinta, jonka avulla voidaan tehdä internetin yli geneerisiä kyselyitä sovitettujen reittien tietokantaan tässä luvussa kuvatuilla hakuehdoilla. Rajapinta toteutettiin Node.js Express-palvelimen avulla. Näiden tiedon julkaisemiseen ja jatkohyödyntämiseen liittyvien palveluiden tarkempi kuvailu on kuitenkin rajattu tämän tutkimuksen ulkopuolelle. Liitteessä 4 on esitetty tilannekuvatietokannasta julkaistu WMS-karttataso, joka on teemoitettu kadun viimeisimmän aurauksen mediaaniaikaleiman ja nykyhetken erotuksen perusteella. Kartan tilanteen päivämäärä on 27.2.2016.

Toteutettujen noin 30 – 60 minuutin viiveellä päivittyvien tietokantataulujen sekä niiden sisältämään tietoon pääsyn antavien REST-, WFS- ja WMS-palveluiden avulla voidaan myöhemmässä vaiheessa toteuttaa halutun kaltainen tilannekuvasovellus katujen kunnossapidosta sekä katujen hoitoon käytettävästä kalustosta. Samaa menetelmää voitaisiin soveltaa myös minkä tahansa muun ajoneuvokategorian toiminnan seuraamiseen.

## 9 Johtopäätökset

Tutkimuksessa toteutetuista algoritmeista suositellaan käytettäväksi sumeiden reittien algoritmia. Algoritmi todettiin sovitustarkkuudeltaan ja suorituskyvyltään paremmaksi kuin verrokina toiminut HMM-karttasovitus, joka esiintyy kirjallisuudessa nykyään yleisimmin käytettynä menetelmänä. Lisäksi sumeiden reittien algoritmi tarjoaa yksinkertaisen ja helposti ymmärrettävän viitekehysten parametrien lisäämiseksi ja muokkaamiseksi mallissa. Sovitustarkkuus autojen osalta osoittautui lähes täysin oikeaksi, kun työkoneiden osalta noin 7 % sovitettujen reittien kokonaispituuksista oli virheellisiä. Työkoneiden liikeratojen karttasovitus virheettömästi tässä tutkimuksessa toteutetuilla ja testatuilla menetelmillä ja käytettävällä datalla osoittautui kuitenkin mahdottomaksi. Karttasovitettua dataa voidaan kuitenkin pitää niin laadukkaana, että sitä voidaan hyödyntää raporteissa, visualisoinneissa ja mahdollisissa tilannekuvasovelluksissa.

On huomioitava, että toisena vaihtoehtona toteutetussa HMM-algoritmissa ei käytetty tässä tutkimuksessa tieverkon attribuutteihin tai ajoneuvon nopeuteen liittyviä parametreja. Mikäli emissio- ja transiitodennäköisyydet mallinnettaisiin useampia muuttujia käyttäen, on todennäköistä, että menetelmän sovitustarkkuus lähestyisi tässä tutkimuksessa esitetyn sumeiden reittien algoritmin sovitustarkkuutta. Tutkimus kuitenkin osoitti, että karttasovitusongelman voi ratkaista tehokkaasti ja korkealla sovitustarkkuudella hyödyntäen HMM:n sijaan sumeaa logiikkaa sekä tavallista graafia ja Dijkstran algoritmia. Tällöin voidaan hyödyntää sumean logiikan etuja eli laskennallista yksinkertaisuutta ja parametrien intuitiivista määrittelyä sumeita syötejoukkoja ja sumeita sääntöjä hyödyntäen.

Ongelmat algoritmien sovitustarkkuudessa johtuvat kolmesta asiasta: paikannuksen epävarmuudesta, matalasta havaintofrekvenssistä tai digitaalisen tieverkon heikosta laadusta. Paikannustarkkuutta voitaisiin Vantaan järjestelmässä pyrkiä parantamaan työkoneiden osalta ottamalla käyttöön DGPS-paikannus, eli hyödyntämällä korjausdataa sijainninmäärittämisessä. Tällöin nykyinen sijaintiepätarkkuus saataisiin todennäköisesti alennettua nykyisestä noin 5 metristä 1-2 metrin tasolle, mikä olisi hyvin todennäköisesti ratkaiseva parannus työkoneiden karttasovitusongelmaan tässä tutkimuksessa käytetyillä algoritmeilla. Tämä vaikuttaisi kustannustehokkaalta ratkaisulta etenkin, mikäli Maanmittauslaitoksen Paikkatietokeskuksen korjausdataa voitaisiin hyödyntää ilmaiseksi.

Toinen mahdollisesti tarkkuutta parantava ratkaisu voisi olla gyroskoopin ja odometrin avulla toteutetun integroidun GNSS/DR-paikannuksen toteutus Kalman-suodinta käyttäen. Tämä lisäisi paikannustarkkuutta todennäköisesti ainakin korkean rakentamisen alueilla ja tunneleiden läheisyydessä. Toisaalta Suomessa ei ole läheskään samankaltaisia korkeasta rakentamista seuraavia ongelmia esimerkiksi satelliittien näkyvyyden tai monitieheijastumien suhteen, kuin joissakin kirjallisuudessa esiintyneissä kaupungeissa, kuten Lontoossa tai Singaporessa. On myös kyseenalaista, kasvattaisiko esimerkiksi gyroskoopilta saatava kulkusuuntatieto merkittävästi paikannuksen tarkkuutta tunnistetuissa ongelmatilanteissa, joissa ajoneuvojen nopeudet ovat melko matalia. Ochieng et al. (2004) huomasivat, että satelliittipaikannukseen liittyvä kulkusuunnan luotettavuuden määrittelyn ongelma alhaisilla nopeuksilla koskee myös heidän testaamaansa integroitua GNSS/DR-paikannusta. Heidän testeissään gyroskoopilla mitatun kulkusuunnan virheet kasvoivat lähes samalla tavalla pienissä nopeuksissa, kuin pelkällä satelliittipaikannuksella määritetyn kulkusuunnankin.

Havaintofrekvenssi toimii Vantaan järjestelmässä tällä hetkellä adaptiivisesti siten, että suurempi hetkellinen kulkusuunnan muutos aiheuttaa lyhyemmän havaintovälin. Havaintofrekvenssin vaikutusta karttasovitukseen voitaisiin testata asettamalla tietyille työkoneille väliaikaisesti tiheämpi havaintoväli. Havaintovälin ja karttasovitustarkkuuden välistä yhteyttä voitaisiin tutkia tekemällä datasta synteettisesti pidempien havaintovälien otantoja ja tutkimalla sovitustarkkuuden muutosta. Näin voitaisiin myös löytää optimaalinen ratkaisu, jossa akku-, tallennus- ja tiedonsiirtokapasiteetti sekä sovitustarkkuus saadaan hyväksyttävälle tasolle.

Käytetty digitaalinen tieverkko eli Liikenneviraston ylläpitämä Digiroad on riittävän laadukasta karttasovitusongelman ratkaisemiseen tässä tutkimuksessa käytetyn datan paikannustarkkuudella. Erityisesti tarkempia paikannusmenetelmiä käytettäessä sijaintitarkkuudeltaan tarkemmin mallinnettu, kattavampi ja ajantasaisempi tieverkko olisi todennäköisesti kuitenkin karttasovituksen kannalta hyödyllinen. Jatkotutkimuksissa kannattaa ehdottomasti testata vuoden 2016 loppuun mennessä valmistuvan Vantaan kaupungin oman keskilinja-aineiston käyttämistä karttasovituksessa. Erityisesti kaistakohtaisen keskilinjamallinnuksen käyttöönotto saattaisi nostaa sovitustarkkuutta merkittävästi, mikäli paikannustarkkuus olisi nykyistä parempi.

Manuaalisesti tehtävässä referenssiaineiston karttasovituksessa havaittiin paikannustarkkuuteen liittyviä haasteita ja paikoitellen esiintyi epävarmuutta oikeasta reitistä. Lisäksi prosessi on melko työläs ja aikaa vievä. Jatkotutkimuksia varten olisi hyödyllistä kerätä referenssi-paikannusdataa jonkin työkoneen todellisista työtehtävistä asentamalla sellaiseen väliaikaisesti tiheää ja tarkkaa paikannusdataa keräävä RTK-VRS-vastaanotin. Tällainen referenssi-aineisto käytännössä eliminoisi epävarmuudet, joita manuaalisessa sovituksessa ilmenee ja se olisi muokattavissa hyvin nopeasti muotoon, jota voitaisiin käyttää hyväksi tässä tutkimuksessa luodussa sovitustarkkuuden vertailuprosessissa.

Mikäli katujen kunnossapidon karttasovitusongelmaan sovellettavia menetelmiä haluttaisiin selvittää uudessa tutkimuksessa, kannattaisi huomio kiinnittää nyt luotujen menetelmien lisäksi datavetoisiin menetelmiin. Sumeiden reittien menetelmä olisi melko helposti muutettavissa Mamdani-tyyppin sumeasta järjestelmästä Sugeno-tyyppin sumeaksi järjestelmäksi. Laadukkaan referenssiaineiston perusteella Sugeno-järjestelmän jäsenyysfunktioiden muodot ja parametrit olisi mahdollista optimoida datavetoisesti neuroverkkoa hyväksikäyttäen. Muista mahdollisista menetelmistä kannattaisi testata esimerkiksi lupaavan oloisia CRF-algoritmeja ja niiden datavetoista optimointia. Uusia muuttujia, joita mahdollisesti voitaisiin käyttää hyväksi, olisi esimerkiksi erilaisten sääolosuhteiden tunnistaminen ja niiden huomiointi ajallisesti, ajoneuvojen tilatiedot sekä säännöt, joita katujen kunnossapitoon liittyy esimerkiksi kunnossapitoluokituksen mukaan. Tällöin karttasovituksessa voitaisiin yrittää ottaa huomioon ennuste ajoneuvojen oletetusta käyttäytymisestä esimerkiksi tietyn suuruisen lumisateen jälkeisellä viikolla.

Suorituskyvyn optimoinnin osalta sumeiden reittien algoritmia kannattaisi kehittää siten, että toteutetaan Koller et al. (2015) esittelemä menetelmä, jossa kandidaattigraafin kaarien painot lasketaan vasta silloin, kun niitä on tarpeen käyttää globaalia optimireittiä haettaessa. Tällöin voidaan jättää kokonaan tekemättä osa tieverkon lyhyimpien reittien hauista sekä sumean järjestelmän evaluoinneista. Koska reittikandidaattien lyhyimpien reittien hakeminen on laskennallisesti raskain operaatio koko algoritmista, voidaan tällä menetelmällä optimoida suorituskyky huomattavasti. Tällöin on myös hyödyllistä vaihtaa optimointiin käytettävä

Dijkstran algoritmi esimerkiksi kaksisuuntaiseksi Dijkstran algoritmiksi, koska silloin optimaalinen reitti voidaan löytää pienemmällä määrällä siirtymisiä graafissa kuin nykyisellä algoritmilla. Toisaalta, mikäli optimointiin voitaisiin käyttää A\*-algoritmia, johon löydetäisiin sovellukseen hyvin sopiva heuristiikka, olisi saavutettavissa vieläkin suurempia hyötyjä.

## 10 Yhteenveto

Tutkimuksen tavoitteena oli kehittää algoritmi Vantaan kaupungin katujen kunnossapitoa tekevien ajoneuvojen telemetria-tiedon karttasovittamiseksi, jotta tietoa voitaisiin hyödyntää tilannekuvasovelluksissa ja tietopalveluissa. Lisäksi haluttiin tietää, onko karttasovitusongelman luotettava ratkaisu mahdollinen käytettävissä olevalla datalla sekä selvittää karttasovituksen virheiden määrä, ja luoda menetelmä virheen suuruuden selvittämiseksi.

Tutkimuksen kirjallisessa osiossa tutustuttiin Vantaan telemetriajärjestelmään sekä yleisemmin liikennetelematiikan menetelmiin ja sovelluksiin. Lisäksi selvitettiin satelliittipaikannusmenetelmien ominaisuuksia sekä perehdyttiin kattavasti erilaisiin kirjallisuudessa toteutettuihin karttasovitusmenetelmiin. Empiirisessä osiossa perehdyttiin tarkemmin Vantaan telemetriajärjestelmän tuottamaan dataan ja kerätyn tiedon perusteella toteutettiin kaksi erilaista karttasovitusalgoritmia, joista uuden tyyppistä sumeisiin reitteihin perustuvaa algoritmia suositeltiin käytettäväksi Vantaan kaupungin telemetria-tiedosta tuotettavien visualisointien, analyysien ja tilannekuvasovellusten karttasovitusmenetelmänä. Lisäksi tutkimuksessa toteutettiin sovitustarkkuuden vertailumenetelmä ja selvitettiin eri algoritmien karttasovituksen tarkkuudet verrattuna referenssiaineistoon. Lopuksi kuvailtiin, miten algoritmi integroitiin osaksi geoprosessointiketjua, jossa data siirtyy rajapinnasta tietokantaan ja siitä edelleen karttasovittuna julkaistavaksi kartta- ja kyselyrajapintoina.

Tutkimuksen perusteella voidaan todeta, että karttasovitusongelma on ratkaistavissa erinomaisella tarkkuudella autojen osalta ja tyydyttävällä tarkkuudella työkonoiden osalta. Työn perusteella käsitys käytössä olevan telemetriajärjestelmän ominaisuuksista ja kyvykkyyksistä sekä sen tuottaman datan hyödyntämisestä kasvoi. Tutkimuksen yhteydessä tehtyjen rajapintojen perusteella on mahdollista käyttää syntyneitä tietoja analyyseissa, uusissa sovelluksissa, julkaista niitä avoimena datana tai tuottaa karttavisualisointeja kaupungin karttapalveluun.

Tässä tutkimuksessa toteutettu sumeiden reittien algoritmi hyödyntää liikeratadataa, joka sisältää havaintojen aikaleimat, kulkusuunnan havainnon hetkellä ja nopeustiedon. Sitä voidaan tutkimuksessa kuvatus sovelluksen lisäksi käyttää minkä tahansa liikeratadatan karttasovitukseen, joka sisältää nämä tiedot ja jonka oletetaan kulkevan tieverkkoa pitkin. Algoritmista on lisäksi helposti muokattavissa yleistetty versio, joka toimii käyttäen hyväkseen pelkkää sijaintitietoa. Tällöin ainoa vaatimus datalle on se, että pisteet syötetään algoritmille oikeassa järjestyksessä.

## 11 Lähdeluettelo

- Alt H., Efrat A., Rote G. & Wenk C. 2003. Matching Planar Maps. Teoksessa: Journal of Algorithms. Vol. 49:2. S. 262-283. Duluth, USA: Academic Press, Inc. ISSN 0196-6774.
- Bazzan, A. 2013. Introduction to Intelligent Systems in Traffic and Transportation. Teoksessa: Synthesis Lectures on Artificial Intelligence and Machine Learning. Vol. 7:3. DOI 10.2200/S00553ED1V01Y201312AIM025.
- Berg, M., Cheong, O., Kreveld, M. & Overmars, M. 2008. Computational Geometry. 3. painos. Berliini, Saksa: Springer. ISBN 978-3-540-77973-5.
- Bernstein, D. & Kornhauser, A. 1996. An Introduction to Map Matching for Personal Navigation Assistants. [Verkkodokumentti]. [Viitattu 5.6.2016]. Saatavissa: <http://www.njtude.org/reports/mapmatchintro.pdf>
- Bishop, C. 1995. Neural Networks for Pattern Recognition. New York, USA: Oxford University Press, Inc. 482 s. ISBN 0198538642.
- Brakatsoulas, S., Pfoser, D., Salas, R. & Wenk, C. 2005. On Map-Matching Vehicle Tracking Data. Teoksessa: Proceedings of the 31st International Conference on Very Large Data Bases. S. 853-864. Trondheim, Norja. ISBN 1-59593-154-6.
- Cassias, I. & Kun, A. 2007. Vehicle telematics: A literature review. University of New Hampshire. [Verkkodokumentti]. [Viitattu 14.6.2016]. Saatavissa: <http://ece-dept-43.unh.edu/Project54/Research/Technical%20Reports/Vehicle%20Telmatrics.pdf>
- Chen D., Diemel A., Guibas, L., Nguyen, A. & Wenk C. 2011. Approximate Map Matching with Respect on the Frechet Distance. Teoksessa: Proceedings of the Meeting on Algorithm Engineering. S. 75-83. ISBN 978-1-61197-291-7.
- Collier, W. 1990. In-Vehicle Route Guidance Systems Using Map Matched Dead Reckoning. Teoksessa: Proceedings of IEEE Position Location and Navigation Symposium. S. 359-363. DOI 10.1109/PLANS.1990.66200.
- Digiroad. 2012. Digiroad laatumalli. [Verkkodokumentti]. [Viitattu 19.6.2016] Saatavissa: <http://www.liikennevirasto.fi/avoindata/palvelut/digiroad/laatu>
- Dijkstra, E.W. 1959. A note on two problems in connexion with graphs. Numerische Mathematik. Vol. 1:1. S. 269-271. DOI 10.1007/BF01386390.
- Durbin, R., Eddy, S., Krogh, A., & Mitchison, G. 1998. Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids. Cambridge, USA: Cambridge University Press. 370 s. ISBN 9780521629713.
- ESNC. 2015. European Satellite Navigation Competition 2015. [Verkkodokumentti]. Saatavissa: <http://www.esnc.eu/index.php?anzeige=prs15.html>



- French, L & Lang, G. 1973. Automatic Route Control System. Teoksessa: IEEE Transactions On Vehicular Technology. Vol. 22:2. S. 36-31. DOI 10.1109/T-VT.1973.23525
- French, L. 1987. Automatic Route Control System. Teoksessa: IEEE Aerospace and Electronic Systems Magazine. Vol. 2:5. S. 6-12. ISSN 0885-8985.
- Fu, L., Sun, D. & Rilett, L. 2006. Heuristic Shortest Path Algorithms for Transportation Applications: State of the Art. Teoksessa: Computers and Operations Research. Vol. 33:11. S. 3324-3343. ISSN 0305-0548.
- Geotrim. Trimnet – uuden satelliittiteknologian hyödyntäminen. [Verkkodokumentti]. [Viitattu 14.7.2016]. Saatavissa: <http://www.geotrim.fi/palvelut/trimnet-vrs>.
- Geodeettinen laitos. Paikannussatelliittijärjestelmät. [Verkkodokumentti]. [Viitattu 21.6.2016]. Saatavissa: <http://www.fgi.fi/fgi/fi/teemat/paikannussatelliittij%C3%A4rjestelm%C3%A4t>
- Goel, A. 2008. Fleet Telematics: Real-Time Management and Planning of Commercial Vehicle Operations. New York, USA: Springer. 184 s. ISBN 978-1-4419-4524-2.
- Goh, C., Dauwels, J., Mitrovic, N., Asif, M., Oran, A. & Jaillet, P. 2012. Online Map-matching Based on Hidden Markov Model for Real-Time Traffic Sensing Applications. Proceedings of the 15th International IEEE Conference on Intelligent Transportation Systems (ITSC). Anchorage, USA. ISBN 978-1-4673-3062-6.
- Greenfeld, J. 2002. Matching GPS Observations to Locations on a Digital Map. Teoksessa: Proceedings of the 81st Annual Meeting of the Transportation Research Board. Washington DC, USA.
- Helsingin Seudun Liikenne. 2016. Ajoneuvoliikenteen teknistoiminnallinen hinnoitteluselvitys. [Verkkodokumentti]. [Viitattu 1.9.2016]. Saatavissa: [https://www.hsl.fi/sites/default/files/hsl\\_julkaisu\\_4\\_2016\\_ajoneuvoliikenteen\\_hinnoitteluselvitys\\_teknistoiminnallinen.pdf](https://www.hsl.fi/sites/default/files/hsl_julkaisu_4_2016_ajoneuvoliikenteen_hinnoitteluselvitys_teknistoiminnallinen.pdf)
- Hummel, B. 2006. Map Matching for Vehicle Guidance. Teoksessa: Drummond, J. & Bilien, R. (toim.) Dynamic and Mobile GIS: Investigating Space and Time. Florida, USA: CRC Press. ISBN 978-1-4200-0860-9.
- Innamaa, S. Malin, F & Rämä, P. 2015. Kilometriverson vaikutukset liikkumiseen. [Verkkodokumentti]. [Viitattu 20.7.2016]. Teknologian tutkimuskeskus VTT, Espoo. Saatavissa: [www.vtt.fi/inf/pdf/technology/2015/T227.pdf](http://www.vtt.fi/inf/pdf/technology/2015/T227.pdf)
- ISO 19157. 2013. Geographic information - Data quality. Geneve, Sveitsi: International Organization for Standardization. 146 s.
- Jagadeesh, G. Srikanthan, T. & Zhang, X. 2004. A Map Matching Method for GPS Based Real-Time Vehicle Location. Teoksessa: Journal of Navigation. Vol 57:3. S. 429-440. ISSN 0373-4633

- JHS 188. 2014. Kansallisen tie- ja katuverkostoaineiston ylläpito ja ylläpitotietojen dokumentointi. Julkisen hallinnon tietohallinnon neuvottelukunta. [Verkkodokumentti]. [Viitattu 28.11.2016]. Saatavissa: <http://www.jhs-suositukset.fi/suomi/jhs188>
- Jurafsky, D & Martin, J. 2014. Speech and Language Processing. Stanford University. [Verkkodokumentti]. [Viitattu 17.9.2016]. Saatavissa: <https://web.stanford.edu/~jurafsky/slp3/8.pdf>.
- Kim, S. & Kim, J. 2001. Adaptive Fuzzy-Network-Based C-Measure Map-Matching Algorithm for Car Navigation System. Teoksessa: IEEE Transactions on Industrial Electronics. Vol. 48:2. S. 432-441. DOI 10.1109/41.915423.
- Kim, W., Jee, G. & Lee, J. 2000. Efficient Use of Digital Road Map In Various Positioning For ITS. Teoksessa: IEEE Symposium on Position Location and Navigation. S. 170-176. San Diego, USA. DOI 10.1109/PLANS.2000.838299.
- Koller, H., Widhalm, P., Dragaschnig, M. & Graser, A. 2015. Fast Hidden Markov Model Map-Matching for Sparse and Noisy Trajectories. IEEE 18th International Conference on Intelligent Transportation Systems. S. 2557-2561. ISBN 978-1-4673-6596-3.
- Krakiwsky, E., Harris, C. & Wong, R. 1988. A Kalman Filter For Integrating Dead Reckoning, Map Matching and GPS Positioning. Teoksessa: Proceedings of IEEE Position Location and Navigation Symposium. S. 39-46. DOI 10.1109/PLANS.1988.195464.
- Kytö, S. 2016. Henkilöautoliikenteen sujuvuuden määrittäminen paikkatietomenetelmin: tapaus pääkaupunkiseutu ja kehyskunnat. Diplomityö. Aalto yliopisto, Espoo.
- L 759/2004. 2004. Laki yksityisyyden suojasta työelämässä. Voimaantulo: 13.8.2004. Finlex. [Viitattu 19.9.2016]. Saatavissa: <http://www.finlex.fi/fi/laki/alkup/2004/20040759>
- Lafferty, J, McCallum, A, Pereira, F. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. Teoksessa: Proceedings of the Eighteenth International Conference on Machine Learning. S. 282-289. San Francisco, USA: Morgan Kaufmann Publishers Inc. ISBN 1-55860-778-1.
- Lakakis, K., Savvaidis, P., Ifadis, M. & Doukas, D. 2004. Quality of Map-matching Procedures Based on DGPS and Stand-Alone GPS Positioning in an Urban Area. Teoksessa: Proceedings of the FIG Working Week. Ateena, Kreikka.
- Langville, A & Meyer, C. 2006. Updating Markov Chains with An Eye On Google's PageRank. Teoksessa: SIAM Journal on Matrix Analysis and Applications. Vol. 27:4. S. 968-987. DOI 10.1137/040619028.
- Laukkanen, A. 2015. Informaatioteknologian hyödyntäminen kaupungin ylläpito- ja rakennustyössä. Diplomityö. Aalto yliopisto, Espoo.

Lehtonen, S. Virtanen, A. & Askola, H. 2016. Liikenteen sähköisten palveluiden tietoturva – niihin kohdistuvat tietoturvariskit ja häirintämenetelmät sekä näiden vaikutukset ja ennaltaehkäisy. Teknologian tutkimuskeskus VTT. ISBN 978-951-38-8406-2.

Leica Geosystems. Leica SmartNet Suomen nykyaikaisin RTK-tukiasemaverkko. [Verkkodokumentti]. [Viitattu 5.10.2016]. Saatavissa: [http://www.leica-geosystems.fi/imagenes/new/common/fi\\_SmartNet\\_Bro.pdf](http://www.leica-geosystems.fi/imagenes/new/common/fi_SmartNet_Bro.pdf).

Li, Y., Huang, Q. & Kerber, M. 2013. Large-Scale Joint Map Matching of GPS Traces. Teoksessa: Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. S. 214–223. New York, USA: ACM. ISBN 978-1-4503-2521-9.

Lou, Y., Zhang, C., Zheng, Y., Xie, X., Wang, W. & Huang, Y. 2009. Map-matching for low-sampling-rate GPS trajectories. Teoksessa: Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (GIS '09). New York, USA: ACM. S. 352-361. ISBN 978-1-60558-649-6.

Maanmittauslaitos. 2016. Maastotietokannan laatumalli. [Verkkodokumentti]. [Viitattu 15.8.2016]. Saatavissa: [http://www.maanmittauslaitos.fi/sites/default/files/Maastotietojen\\_laatumalli.pdf](http://www.maanmittauslaitos.fi/sites/default/files/Maastotietojen_laatumalli.pdf).

Manning, C. & Schütze, H. 1999. Foundations of Statistical Natural Language Processing. Cambridge, USA: MIT Press. 720 s. ISBN 9780262312127.

MathWorks. 2016. Fuzzy logic toolbox user's guide. The Mathworks inc. [Verkkodokumentti]. Saatavissa: [http://www.mathworks.com/help/pdf\\_doc/fuzzy/fuzzy.pdf](http://www.mathworks.com/help/pdf_doc/fuzzy/fuzzy.pdf).

McBratney, A. & Odeh, I. 1997. Application of Fuzzy Sets in Soil Science: Fuzzy Logic, Fuzzy Measurements and Fuzzy Decisions. Teoksessa: Geoderma. Vol. 77:2-4. S. 85-113. DOI 10.1016/S0016-7061(97)00017-7.

Mehtälä, I. 2012. Yhden vastaanottimen tarkka paikannus. [Verkkodokumentti]. [Viitattu: 19.8.2016]. Opinnäytetyö. Metropolia ammattikorkeakoulu, Espoo. Saatavissa: <https://www.theseus.fi/handle/10024/43254>

Miller, H & Han, J. 2001. Geographic Data Mining and Knowledge Discovery. Bristol, USA: Taylor & Francis. ISBN 0415233690.

MIT. Big O notation. Massachusetts Institute of Technology. [Verkkodokumentti]. [Viitattu 2.7.2016]. Saatavissa: [http://web.mit.edu/16.070/www/lecture/big\\_o.pdf](http://web.mit.edu/16.070/www/lecture/big_o.pdf)

MML Paikkatietokeskus. DGNSS-palvelu. [Verkkodokumentti]. [Viitattu: 20.10.2016]. Saatavissa: <http://euref-fin.fgi.fi/fgi/fi/paikannuspalvelu/dgnss-palvelu>

Najjar, M. & Bonnifait, P. 2003. A Roadmap Matching Method For Precise Vehicle Localization Using Belief Theory And Kalman Filtering. Teoksessa: The 11th International Conference in Advanced Robotics. Coimbra, Portugal.

Newson, P. & Krumm, J. 2009. Hidden Markov Map Matching Through Noise and Sparseness. Teoksessa: Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (GIS '09). New York, USA: ACM. S. 336–343. ISBN 978-1-60558-649-6.

Ochieng, W., Quddus, M. & Noland, R. 2004. Integrated positioning algorithms for transport telematics applications. Proceedings of the 17th International Technical Meeting of the Satellite Division of the Institute of Navigation. California, USA. S. 692-705.

Ollila et al. 2013. Oikeudenmukaista ja älykästä liikennettä, työryhmän loppuraportti. [Verkkodokumentti]. [Viitattu 25.7.2016] Saatavissa: <http://www.lvm.fi/-/oikeudenmukaista-ja-alykasta-liikennetta-tyoryhman-loppuraportti-810727>. ISBN 978-952-243-372-5.

Osogami, T & Raymond, R. 2013. Map Matching with Inverse Reinforcement Learning. Teoksessa: Proceedings of the Twenty-Third international joint conference on Artificial Intelligence (IJCAI '13). Rossi, F. (toim.) Palo Alto, USA: AAAI Press. S. 2547-2553. ISBN 978-1-57735-633-2.

PostgreSQL Global Development Group. 2016. PostgreSQL 9.6rc1 Documentation. [Verkkodokumentti]. [Viitattu 12.9.2016] Saatavissa: <https://www.postgresql.org/docs/9.6/static/index.html>.

Poutanen, M. 2016. Satelliittipaikannus. Helsinki, Suomi: Ursa. 349 s. ISBN 978-952-5985-41-2.

Quddus, M., Ochieng, W., Zhao, L. & Noland, R. 2003. A General Map Matching Algorithm for Transport Telematics Applications. GPS Solutions. Vol. 7:3. S. 157-167.

Quddus, M., Ochieng, W. & Noland, R. 2006. A High Accuracy Fuzzy Logic Based Map Matching Algorithm for Road Transport. Journal of Intelligent Transportation Systems. Vol. 10:3. S. 103-115. DOI 10.1080/15472450600793560.

Quddus, M. 2006. High Integrity Map Matching Algorithms for Advanced Transport Telematics Applications. Väitöskirja. Centre for Transport Studies, Department of Civil and Environmental Engineering, Imperial College London, United Kingdom.

Quddus, M. & Ochieng, W. 2014. Shortest Path and Vehicle Trajectory Aided Map-Matching for low Frequency GPS Data. Transportation Research Part C: Emerging Technologies. Vol. 55. S. 328-339.

Rabiner, L. 1990. A Tutorial On Hidden Markov Models and Selected Applications in Speech Recognition. Teoksessa: Waibel, A & Lee, K. (toim.) Readings in speech recognition. San Francisco, USA: Morgan Kaufmann Publishers Inc. S. 267-296. ISBN 1-55860-124-4.

R, Wilson. 1996. Introduction to Graph Theory. 4. painos. Harlow, Iso-Britannia: Longman Group. 171 s. ISBN 0-582-24993-7.

R Archive Network. 2015. [Verkkodokumentti]. [Viitattu 26.10.2016]. Saatavissa: <https://cran.r-project.org/web/packages/anfis/anfis.pdf>.

Russell, S & Norvig, P. 2009. Artificial Intelligence: A Modern Approach. 3. painos. New Jersey, USA: Prentice Hall. 1152 s. ISBN 0-13-604259-7.

Safe Software. 2016. FME Knowledge Center. [Verkkodokumentti]. [Viitattu 8.8.2016]. Saatavissa: <https://support.safe.com/knowledgedocumentation>.

Said, H., Nicoletti, T. & Perez-Hernandez, P. 2014. Utilizing Telematics Data to Support Effective Equipment Fleet-Management Decisions: Utilization Rate and Hazard Functions. Teoksessa: Journal of Computing in Civil Engineering. Vol. 30:1. ISSN 1943-5487.

Scott, C. 1994. Improved GPS Positioning for Motor Vehicles Through Map Matching. Teoksessa: Proceedings of the Institute of Navigation (ION) Annual Conference. Salt Lake City, Utah. S. 1391-1400.

Seeber, G. 2003. Satellite Geodesy: Foundations, Methods, and Applications. 2. painos. New York, USA: De Gruyter. 589 s. ISBN 978-3110175493.

Song, R., Lu, W. & Sun, W. 2012. Quick Map Matching Using Multi-Core CPUs. Proceedings of the 20th International Conference on Advances in Geographic Information Systems. New York, USA: ACM. ISBN 978-1-4503-1691-0.

Srinivasan, D. & Cheu, R. 2003. Development of an Improved ERP System Using GPS and AI Techniques. Teoksessa: IEEE Proceedings on Intelligent Transportation Systems. S. 554–559. DOI 10.1109/ITSC.2003.1252014.

Syed, S. & Cannon, M. 2004. Fuzzy Logic-Based Map Matching Algorithm For Vehicle Navigation System In Urban Canyons. Teoksessa: Proceedings of the Institute of Navigation (ION) National Technical Meeting. San Diego, USA. S. 26–28. DOI 10.1.1.541.6810.

Szwed, P & Pekala, K. 2014. An Incremental Map-Matching Algorithm Based on Hidden Markov Model. Artificial Intelligence and Soft Computing: 13th International Conference. Vol. 8468. Springer. S. 579-590. ISBN 978-3-319-07176-3.

Takagi, T & Sugeno, M. 1985. Fuzzy identification of systems and its applications to modeling and control. IEEE Transactions on Systems, Man, and Cybernetics. Vol. 15:1. S. 116-132. Saatavissa: <http://ieeexplore.ieee.org/document/6313399>. ISSN 0018-9472.

Tanaka, J. 1990. Navigation System with Map-Matching Method. Proceedings of the SAE International Congress and Exposition. S. 45-50. DOI 10.4271/900471.

Taylor, G., Blewitt, G., Steup, D., Corbett, S. & Car, A. 2001. Road Reduction Filtering for GPS–GIS Navigation. Transactions In GIS. Vol. 5:3. S. 193–207. ISSN 1361-1682.

Thede, S. 2004. An Introduction to Genetic Algorithms. Journal of Computing Sciences in Colleges. Vol. 20:1. USA. Consortium for Computing Sciences in Colleges. S. 115-123. ISSN 1937-4771.

Tiittanen, E. 2016. Helsingin kaupungin rakentamispalvelu Stara, katujen kunnossapito. Maansiirtokoneen kuljettaja. Haastattelu 16.7.2016.

Vantaan kaupunki. 2012. Ajoneuvojen paikantamisen käytösäännöt. 4.12.2012.

Vantaan kaupunki. Katujen kunnossapito. [Verkkodokumentti]. [Viitattu 7.8.2016]. Saatavissa: [http://www.vantaa.fi/asuminen\\_ja\\_ymparisto/kadut\\_ja\\_viheralueet/kadut/katujen\\_kunnossapito](http://www.vantaa.fi/asuminen_ja_ymparisto/kadut_ja_viheralueet/kadut/katujen_kunnossapito).

Velaga, N., Quddus, M. & Bristow, L. 2009. Developing an enhanced weight-based topological map-matching algorithm for intelligent transport systems. *Transportation Research Part C: Emerging Technologies*. Vol. 17:6. ISSN 0968-090X.

Vättö, J. 2015. Haastattelu vantaa.fi verkkosivuilla. [Verkkodokumentti]. [Viitattu 14.9.2016]. Saatavissa: [http://www.vantaa.fi/uutisia/kaikki\\_uutiset/101/0/120455](http://www.vantaa.fi/uutisia/kaikki_uutiset/101/0/120455)

Wei, H., Wang Y. & Foreman, G. 2012. Fast Viterbi Map Matching with Tunable Weight Functions. *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*. New York, USA: ACM. ISBN 978-1-4503-1691-0.

Wen, Y. 2007. Using Log-transform to Avoid Underflow Problem in Computing Posterior Probabilities. Massachusetts Institute of Technology. [Verkkodokumentti]. [Viitattu 5.10.2016]. Saatavissa: [http://web.mit.edu/wenyang/www/log\\_transform\\_for\\_underflow.pdf](http://web.mit.edu/wenyang/www/log_transform_for_underflow.pdf).

White, C., Bernstein, D. & Kornhauser, A. 2000. Some Map Matching Algorithms for Personal Navigation Assistants. *Transportation Research Part C: Emerging Technologies*. Vol. 8:1-6. S. 91–108. DOI 10.1016/S0968-090X(00)00026-7.

Xu, Y. 2000. Development of Transport Telematics in Europe. *Geoinformatica*. Vol. 4:2. S. 179–200. DOI 10.1023/A:1009876223822.

Yang, J. & Meng, L. 2014. Feature Selection in Conditional Random Fields for Map Matching of GPS Trajectories. Teoksessa: *Lecture Notes in Geoinformation and Cartography*. S. 121-135. ISSN 1863-2246.

Yhdysvaltain puolustusministeriö. 2008. Global Positioning System Standard Positioning Service Performance Standard. Assistant secretary of defense for command, control, communications, and intelligence.

Yuan, J., Zheng, Y., Zhang, C., Xie, X., & Sun, G. 2010. An Interactive-Voting Based Map Matching Algorithm. *Eleventh International Conference on Mobile Data Management*. S. 43–52. ISBN 978-0-7695-4048-1.

Zadeh, L. 1973. Outline of a new approach to the analysis of complex systems and decision processes. Teoksessa: *IEEE Transactions on Systems, Man, and Cybernetics*. Vol. 3:1. S. 28-44. ISSN 2168-2909.

Zadeh, L. 1965. Fuzzy sets. *Teoksessa: Information and Control*. Vol. 8:3. Berkeley, USA: University of California, Berkeley. S. 338-353. DOI 10.1016/S0019-9958(65)90241-X.

Zhang, J. & Goodchild, M. 2002. *Uncertainty in Geographical Information*. New York, USA: Taylor and Francis. 266 s. ISBN 0-203-47132-6.

Zhao, Y. 1997. *Vehicle Location and Navigation System*. Montana, USA: Artech House Inc. 345 s. ISBN 0890068615.

Zhao, L., Ochieng, W., Quddus, M. & Noland, R. 2003. An Extended Kalman Filter Algorithm for Integrating GPS and Low Cost Dead Reckoning System Data for Vehicle Performance and Emissions Monitoring. *Journal of Navigation*. Vol. 56:2. S. 257–275. DOI 10.1017/S0373463303002212.

## **Liiteluettelo**

Liite 1 Vantaan kaupungin talvikunnossapitokalustoa

Liite 2 Järjestelmän tietokantataulujen kuvaus

Liite 3 Prosessin kuvaus

Liite 4 WMS-taso katujen avaruuden tilannekuvataulusta



## Liite 1 Vantaan kaupungin talvikunnossapidokalustoa

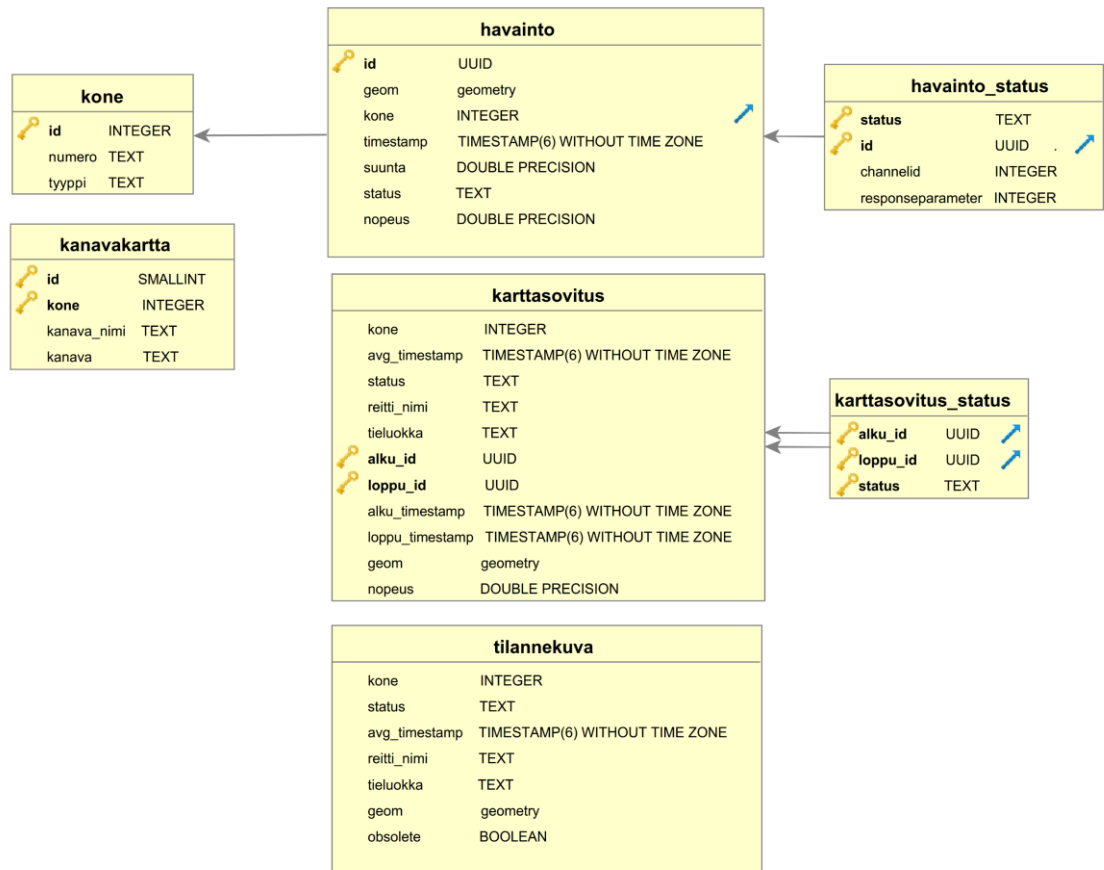


*Volvon kuorma-auto etu- ja keskiauralla.*

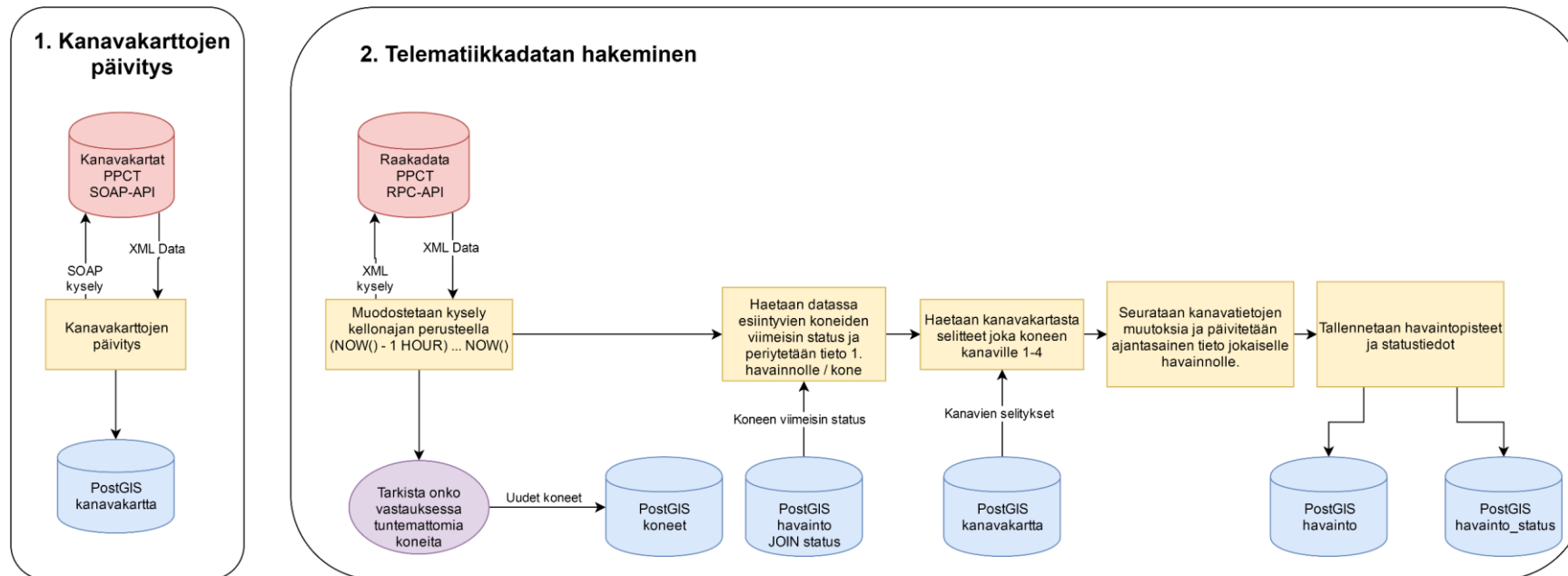


*Wille pyöräkuormaaja etuauralla ja pesulaitteella.*

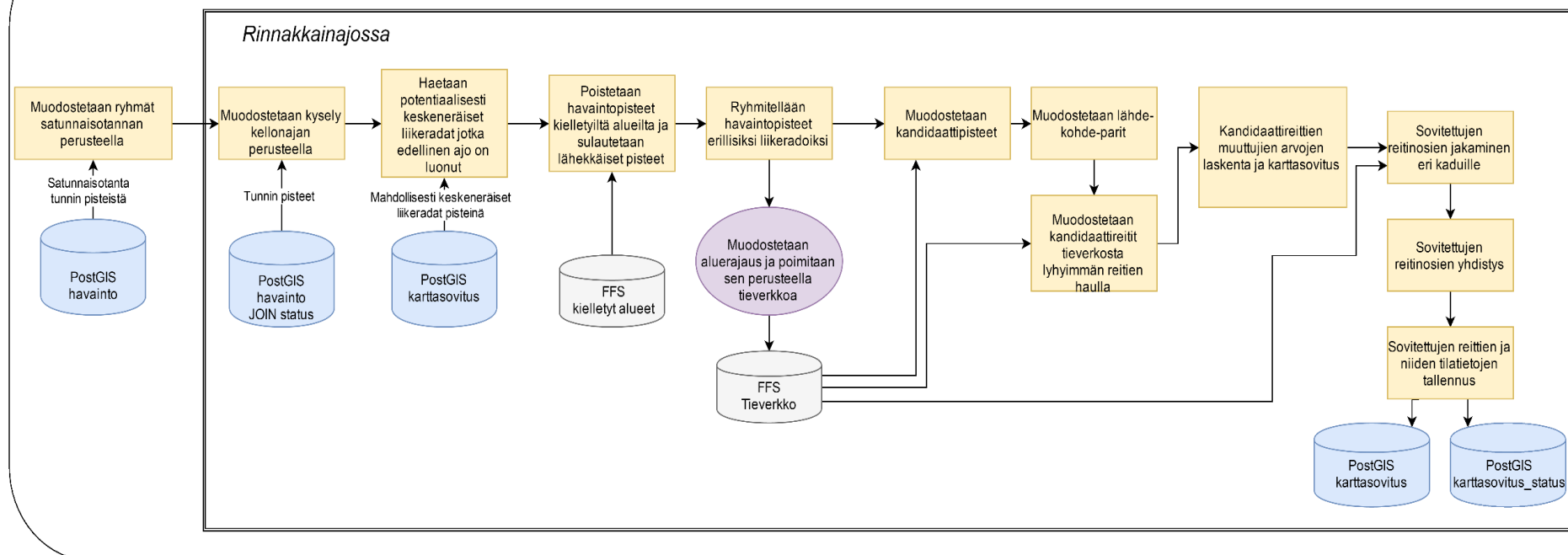
## Liite 2 Järjestelmän tietokantataulujen kuvaus



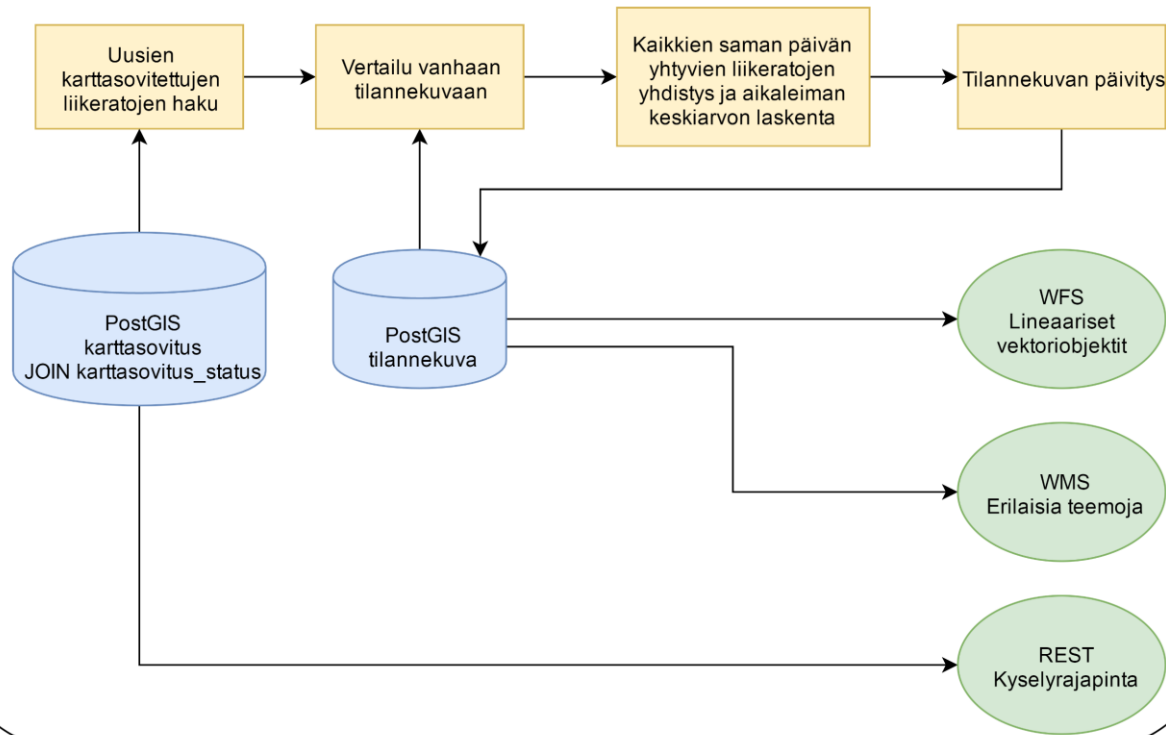
## Liite 3 Prosessin kuvaus



## 3. Karttasovitus



#### 4. Tilannekuvan päivitys ja datan julkaisukanavat





## Liite 4 WMS-taso katujen aurauksen tilannekuvataulusta

